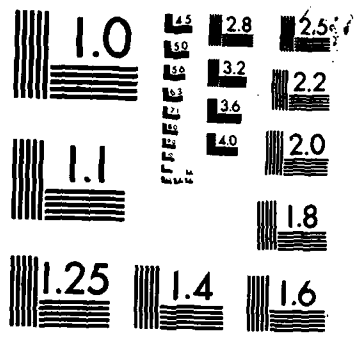
AD-A141 486   LCAP2 (LINEARCCONTROL ANALYSIS PROGRAM) VOLUME 1 BATCH   1/2
LCAP2 USER'S GUIDE..(U) AEROSPACE CORP EL SEGUNDO CA
GUIDANCE AND CONTROL DIV   E A LEE 15 NOV 83
UNCLASSIFIED   SD-TR-84-06-VOL-1 F04701-83-C-0084   F/G 9/2   NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

REPORT SD-TR-84-06, Vol. I

(12)

LCAP2 - Linear Control Analysis Program

Volume I: Batch LCAP2 User's Guide

E. A. LEE
Guidance and Control Division
Engineering Group
The Aerospace Corporation
El Segundo, Calif. 90245

15 November 1983

Final Report

APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED

Prepared for

SPACE DIVISION
AIR FORCE SYSTEMS COMMAND
Los Angeles Air Force Station
P.O. Box 92960, Worldway Postal Center
Los Angeles, California 90009

This report was submitted by The Aerospace Corporation, El Segundo, CA 90245, under Contract No. F04701-83-C-0084 with the Space Division, P.O. Box 92960, Worldway Postal Center, Los Angeles, CA 90009. It was reviewed and approved for The Aerospace Corporation by Ronald G. Nishinaga, Advanced Programs, Defense Development Division. Major William J. Cooper, SD/YDS, was the Air Force project officer.

This report has been reviewed by the Public Affairs Office (PAS) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nationals.

This technical report has been reviewed and is approved for publication. Publication of this report does not constitute Air Force approval of the report's findings or conclusions. It is published only for the exchange and stimulation of ideas.

William J. Cooper
Project Officer

Joseph Hess
Joseph Hess, GM-15, Director
West Coast Office, Air Force
Space Technology Center

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER SD-TR-84-06, Vol. I	2. GOVT ACCESSION NO. AD-A141486	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) LCAP2 - LINEAR CONTROL ANALYSIS PROGRAM Volume I: Batch LCAP2 User's Guide		5. TYPE OF REPORT & PERIOD COVERED Final Report 6. PERFORMING ORG. REPORT NUMBER TR-0084(9975)-1, Vol. I
7. AUTHOR(s) Eugene A. Lee		8. CONTRACT OR GRANT NUMBER(s) F04701-83-C-0084
9. PERFORMING ORGANIZATION NAME AND ADDRESS The Aerospace Corporation El Segundo, Calif. 90245		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Space Division Los Angeles Air Force Station Los Angeles, Calif. 90009		12. REPORT DATE 15 November 1983 13. NUMBER OF PAGES 178
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Linear systems	Sampled data
Digital control systems	z-transform
Frequency response	w-transform
Root locus	Multirate sampling
Laplace transform	Frequency decomposition

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The computer program LCAP2 (Linear Controls Analysis Program) provides the analyst with the capability to numerically perform classical linear control analysis techniques such as transfer function manipulation, transfer function evaluation, frequency response, root locus, time response and sampled-data transforms. It is able to deal with continuous and sampled-data systems, including multiloop multirate digital systems, using s, z and w transforms. over

DD FORM 1473 (FACSIMILE)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

19. KEY WORDS (Continued)

Control systems analysis program
Inverse transforms
Cramer's method
Transfer function evaluation

20. ABSTRACT (Continued)

Primary considerations in the development of this program were ease of use and computational accuracy. Transfer function and polynomial arrays are defined to be referenced with indices so that they may be easily addressed by the operators. The combination of this set of LCAP2 operators and the form of the data structure provides a very flexible and easy to use program.

Since each LCAP2 operator is coded as a FORTRAN subroutine, the batch version of LCAP2 allows the user to easily develop code to automate, for example, a complete stability analysis task beginning with the input of raw data to the generation of the stability plots. An interactive version of LCAP2 is also available.

The LCAP2 report is organized in three volumes: batch user's guide (I), interactive user's guide (II), and source code description (III).

PREFACE

This is the first formal issue of the user's guide for Batch LCAP2. Numerous examples describing the more common LCAP2 operators are presented.

Batch LCAP2 is an improved version of LCAP, Ref. 1, which was originally developed in 1966. The major difference in usage between these two programs is the FORTRAN callable implementation of each LCAP2 operator which enables the analyst to develop code to automate the analysis of complex systems.

CONTENTS

1.0 INTRODUCTION

Batch LCAP2 (Linear Controls Analysis Program) is a FORTRAN program which provides the control analyst with the capability to numerically perform classical linear control analysis techniques such as transfer function manipulation, transfer function evaluation, frequency response, root locus, inverse time response and sampled-data transforms. It is able to deal with continuous and sampled-data systems, including multiloop multirate digital systems, using s, z and w transforms.

2.0 DESCRIPTION

This program was designed to provide the control system analyst with most of the classical analysis tools needed for analyzing continuous and sampled-data systems by transform techniques. A set of transfer function and polynomial operators has been defined in a fashion similar to the instruction set of a computer. Transfer function and polynomial arrays are defined to be referenced with indices so that they may be easily addressed by the operators. The combination of this set of LCAP2 operators and the form of the data structure provides a very flexible and easy to use program.

The data structure of the program includes (1) s, z and w plane transfer functions designated as SPTFi, i=1,2..., ZPTFi, i=1,2..., WPTFi i=1,2..., respectively, and (2) polynomials designated as POLYi ,i=1,2... Operations on these transfer functions or polynomials are specified by references to their indices. An arbitrarily large number of transfer functions and polynomials are available to the user since disk storage is utilized when the number becomes too large.

The transfer functions are represented as ratios of polynomials. The user can load data into a transfer function using either the coefficient or the root form representation. Data structures used for the transfer functions and polynomials require that the order of the polynomials be less than fifty.

Each of the LCAP2 operators is implemented as a FORTRAN subroutine with a minimal number of arguments used to specify the transfer functions or polynomials involved. For example, to add SPTF1 to SPTF2 and store the results into SPTF3, the FORTRAN statement is CALL SPADD(3,1,2).

A typical use of these operators for a simple system would be to reduce a block diagram to a single open or closed loop transfer function using the add, subtract, multiply and divide operators. Then one of the operators used to implement the classical control analysis techniques can be applied. For example, if SPTFi is the open loop transfer function, the operator SFREQ(i) can be used to compute the frequency response so that the system can be evaluated.

For complex continuous systems, the above block diagram reduction method may not be practical to apply. Cramer's method for transfer function evaluation could then be used.

For sampled-data systems z and w plane operators are provided. Analysis of small order systems can be performed in either the z or w plane. The analyst may prefer to perform the analysis in the w plane since this would allow the use of the Bode design techniques. However, if the order of the system is high, the analysis must be performed in the w plane since z plane coefficients cannot be as accurately represented by the computer.

Multirate sampled-data systems with integer rate sampling can be analyzed by LCAP2. Two types of operators based upon Sklansky's frequency decomposition method are available for this type of analysis.

3.0 LANGUAGE AND HOST COMPUTER

The program is written entirely in CDC (Control Data Corp.) FORTRAN EXTENDED 4 with the exception of one subroutine which is written in assembly language. The batch version of LCAP2 runs on the CDC 176 under the SCOPE 2.1 Operating system. Batch jobs typically require 140-240k words. Description of the code for this program is given in Ref. 2.

An interactive version of LCAP2, Ref. 3, is available using the CDC INTERCOM which runs on the CDC 835 computer. INTERCOM can also be used to define and edit a batch file which can be submitted to the CDC 176.

4.0 JOB STRUCTURE

The basic operations of a batch LCAP2 job are:

(1) Creation of FORTRAN program - main program and subroutines (optional)

(2) Compilation of the source code from (1)

(3) Loading of routines from the LCAP2 and system libraries

(4) Execution of the LCAP2 program from (2)

(5) (Optional) - Cataloging of data file if one is created by LCAP2

(6) (Optional) - Loading and execution of the HARDCPY program to produce hardcopy plots created by LCAP2

To facilitate the development of the FORTRAN program by the user, the CDC UPDATE[1] program is utilized. An LCAP2 program library has been defined so that the first part of the main program, which contains many lines of COMMON block and EQUIVALENCE statements, need not be written by the user. This block of code is copied from the program library and added to the the user's FORTRAN code to create the source code. The input (card images) for the UPDATE program will be of the form:

```
*IDENT idname
*INSERT START.1
*DECK MAIN
*CALL LCAP2
      CALL INITO        (initialization of LCAP2 parameters)
      CALL MINITO       (initialization of matrix parameters)
           .
           .
      (user's FORTRAN code)
           .
           .
      CALL LEXIT        (required if hardcopy plots are generated)
      END
```

The * in column 1 defines an UPDATE directive. The first directive, *IDENT idname, specifies an identification name, idname, which can be 1 through 9 characters long. The second directive, *INSERT START.1, defines the location where the input data to follow is to be inserted. The directive, *CALL LCAP2, will write the code in COMDECK LCAP2 to the file COMPILE. This COMDECK LCAP2 contains the main program statment and all of the COMMON block and EQUIVALENCE statements required by the main program. The remaining input data are the user's FORTRAN code which will be copied to the file COMPILE to complete the creation of the main program. The words in parenthesis are comments and are not part of the FORTRAN code.

The job control cards for setting up the above operations are given in the next section.

---

[1] The UPDATE program maintains and updates source decks for libraries under the SCOPE 2, NOS 1, and NOS/BE 1 operating systems.

5.0 JOB SUBMITTAL

Two forms for job submittal are given below. The first will be an explicit one which includes a complete list of control cards required. The second is a shortened form which attaches and uses a procedure to generate the control cards.

The *first form* is:

```
                  .
(    control cards for accounting   )
                  .
FILE,TAPE30,BT=I.                            (optional, use only if old
ATTACH(TAPE30,lfn,ID=.....,ST=PF6)            data is to be restored)
ATTACH(OLDPL,8LCAP2PLX,ID=9487)              (attach LCAP2 program library)
UPDATE.
FTN(I=COMPILE,R=3)                           (compile output of UPDATE)
FILE,TAPE31,BT=I.                            (optional, only if LCAP2
REQUEST(TAPE31,*PF)                           STORE operator is to be used)
RETURN(OLDPL)
ATTACH(LCAPLIB,8LCAP2LIBX,ID=9487)           (attach LCAP2 library)
ATTACH(PLOTLIB,3FTNPLOTLIB)                  (attach plot library)
LIBRARY(LCAPLIB,PLOTLIB)
LDSET(PRESET=ZERO)
LGO.                                         (load and execute LCAP2 program)
CATALOG(TAPE31,8filename,ID=.....,ST=PF6)    (optional, use only if LCAP2
                                              STORE operator was used)
HARDCPY,ST=IBMD8.                            (omit argument if A3 plotter desired)
*EOR                                         (end of record)
               .
               .
  (UPDATE input deck as described
   in previous section)
               .
               .
*EOR                                (end of record)
```

The **second form** is :

```
--------------------------------------------------------------------------------
                .
(    control cards for accounting    )
                .
FILE,TAPE30,BY=I.                             (optional, use only if old
ATTACH(TAPE30,lfn,ID=.....,ST=PF6)             data is to be restored)
ATTACH(X,8LCAP2CC,ID=9487)                    (attach LCAP2 control card procedure)
BEGIN,LCAP2CC,X.
CATALOG(TAPE31,8filename,ID=.....,ST=PF6)     (optional, use only if LCAP2
                                               STORE operator was used)
HARDCPY,ST=IBMD8.                             (omit argument if A3 plotter desired)
*EOR                                          (end of record)
              .
              .
  (UPDATE input deck as described
   in previous section)
              .
              .
*EOR                                          (end of record)
--------------------------------------------------------------------------------
```

In the second form, the file X will generate the same control cards as the first form except for the (1) FILE,TAPE30,..., (2) ATTACH(TAPE30,..., (3) CATALOG(TAPE31,... , and (4) HARDCPY statements. It is recommended that the second form be used unless the user must change some of the control cards. An example when this is necessary is if the print limit is exceeded. The statement LGO. should be changed to LGO(PL=.....) where the value of PL is the number of print lines.

6.0 LIST OF LCAP2 OPERATORS

The following is a list of the LCAP2 operators grouped by type of operation. A brief description of each operation is included. A more detailed description of each operator is given in Appendix A.

POLYNOMIAL OPERATORS

```
PADD(i,j,k)   - Polynomial Add
                POLYi = POLYj + POLYk

PEQU(i,j)     - Polynomial Equal
                POLYi = POLYj

PLDC(i)       - Polynomial Load, Coefficient Form
                POLYi = POLY

PLDR(i)       - Polynomial Load, Root Form
                ROOTi = ROOT,  POLYi = PSYNTH(ROOTi)

PMPY(i,j,k)   - Polynomial Multiply
                POLYi = POLYj * POLYk

PPRN(i)       - Polynomial Print
                Print out POLYi

PRTS(i)       - Find Roots Of Polynomial
                ROOTi = Roots of POLYi

PSUB(i,j,k)   - Polynomial Subtract
                POLYi = POLYj - POLYk
```

S-PLANE OPERATORS

```
CPYPS(i,j,k) - Copy Polynomials Into S-Plane Transfer Function
               SPTFi = POLYj / POLYk

CPYSP(i,j,k) - Copy S-Plane Transfer Function Into Polynomials
               POLYj = numerator of SPTFi
               POLYk = denominator of SPTFi

FREQS(FAUX1) - S-Plane Frequency Response With User Supplied Function

SELCR(i)     - Eliminate Common Roots From S-Plane Transfer Function
               SPTFi = SPTFi with common roots eliminated
```

SFREQ(i) - S-Plane Frequency Response
Compute frequency response of SPTFi

SLOCI(i) - S-Plane Root Locus
Compute root locus of SPTFi

SNORM(i) - Normalize S-Plane Transfer Function
SPTFi = SPTFi with coefficients normalized

SPADD(i,j,k) - S-Plane Transfer Function Add
SPTFi = SPTFj + SPTFk

SPDIV(i,j,k) - S-Plane Transfer Function Divide
SPTFi = SPTFj / SPTFk

SPEQU(i,j) - S-Plane Transfer Function Equal
SPTFi = SPTFj

SPLDC(i) - S-Plane Transfer Function Load, Coefficient Form
SPTFi = POLYN / POLYD

SPLDR(i) - S-Plane Transfer Function Load, Root Form
SROOTi = ROOTN / ROOTD , SPTFi = PSYNTH(SROOTi)

SPMPY(i,j,k) - S-Plane Transfer Function Multiply
SPFTi = SPTFj * SPTFk

SPPRN(i) - S-Plane Transfer Function Print
Print out SPTFi

SPRTS(i) - Find Roots Of S-Plane Transfer Function
SROOTi = Roots of SPTFi

SPSUB(i,j,k) - S-Plane Transfer Function Subtract
SPTFi = SPTFj - SPTFk

STIME(i) - Inverse Laplace Transform and Time Response
Compute time response of SPTFi by partial fraction expansion

SWMRX(i,j) - S-to-W Plane Multirate Transform
(slow-to-fast sampler)
WPTFi = w plane multirate transform of SPTFj

SWXFM(i,j) - S-to-W Plane Transform
WPTFi = z plane transform of SPTFj

SZMRX(i,j) - S-to-Z Plane Multirate Transform
(slow-to-fast sampler)
ZPTFi = z plane multirate transform of SPTFj

SZXFM(i,j) - S-to-Z Plane Transform
ZPTFi = z plane transfrom of ?TFj

Z-PLANE OPERATORS

CPYPZ(i,j,k) - Copy Polynomials Into Z-Plane Transfer Function
ZPTFi = POLYj / POLYk

CPYZP(i,j,k) - Copy Z-Plane Transfer Function Into Polynomials
POLYj = numerator of ZPTFi
POLYk = denominator of ZPTFi

FREQZ(FAUX1) - Z-Plane Frequency Response With User Supplied Function

ZELCR(i) - Eliminate Common Roots From Z-Plane Transfer Function
ZPTFi = ZPTFi with common roots eliminated

ZFREQ(i) - Z-Plane Frequency Response
Compute frequency response of ZPTFi

ZLOCI(i) - Z-Plane Root Locus
Compute root locus of ZPTFi

ZMRFQ(i) - Z-Plane Multirate Frequency Response
Compute multirate rate frequency response of ZPTFi
by application of frequency decomposition method

ZMRXFM(i,j) Z-Plane Multirate Transform By Frequency Decomposition
(fast-to-slow sampler)
ZPTFi = Multirate transform of ZPTFj by application
of frequency decomposition method.

ZNORM(i) - Normalize Z-Plane Transfer Function
ZPTFi = ZPTFj with coefficients normalized

ZPADD(i,j,k) - Z-Plane Transfer Function Add
ZPTFi = ZPTFj + ZPTFk

ZPDIV(i,j,k) - Z-Plane Transfer Function Divide
ZPTFi = ZPTFj / ZPTFk

ZPEQU(i,j) - Z-Plane Transfer Function Equal
ZPTFi = ZPTFj

ZPLDC(i) - Z-Plane Transfer Function Load, Coefficient Form
ZPTFi = POLYN / POLYD

ZPLDR(i) - Z-Plane Transfer Function Load, Root Form
ZROOTi = ROOTN/ ROOTD, ZPTFi = PSYNTH(ZROOTi)

ZPMPY(i,j,k) - Z-Plane Transfer Function Multiply
ZPTFi = ZPTFj * ZPTFk

ZPPRN(i) - Z-Plane Transfer Function Print
Print out ZPTFi

ZPRTS(i) - Find Roots Of Z-Plane Transfer Function
ZROOTi = Roots of ZPTFi

ZPSUB(i,j,k) - Z-Plane Transfer Function Subtract
ZPTFi = ZPTFj - ZPTFk

ZSXFM(i) - Z-to-S Root Transformation
Compute "s plane equivalent" of roots of ZPTFi

ZTIME(i) - Inverse Z-Transform and Time Response
Compute time response of ZPTFi

ZVCNG(i,j,n) - Z-to-ZN Transform
ZPTFi = ZPTFj with z replaced with z**n

ZWXFM(i,j) - Z-to-W Plane Transform
WPTFi = Bilinear transform of ZPTFi

<u>W-PLANE OPERATORS</u>

CPYPW(i,j,k) - Copy Polynomials Into W-Plane Transfer Function
WPTFi = POLYj / POLYk

CPYWP(i,j,k) - Copy W-Plane Transfer Function Into Polynomials
POLYj = numerator of WPTFi
POLYk = denominator of WPTFi

FREQZ(FAUX1) - Z-Plane Frequency Response With User Supplied Function

WELCR(i) - Eliminate Common Roots From W-Plane Transfer Function
WPTFi = WPTFi with common roots eliminated

WFREQ(i) - W-Plane Frequency Response
Compute frequency response of WPTFi

WLOCI(i) - W-Plane Root Locus
Compute root locus of WPTFi

WMRFQ(i) - W-Plane Multirate Frequency Response
Compute multirate frequency response of WPTFi by application of frequency decomposition method

WMRXFM(i,j) - W-Plane Multirate Transform By Frequency Decomposition (fast-to-slow sampler)
WPTFi = Multirate transform of WPTFj by application frequency decomposition method

WNORM(i) - Normalize W-Plane Transfer Function
WPTFi = WPTFi with coefficients normalized

WPADD(i,j,k) - W-Plane Transfer Function Add
WPTFi = WPTFj + WPTFk

WPDIV(i,j,k) - W-Plane Transfer Function Divide
WPTFi = WPTFj / WPTFk

WPEQU(i,j) - W-Plane Transfer Function Equal
WPTFi = WPTFj

WPLDC(i) - W-Plane Transfer Function Load, Coefficient Form
WPTFi = POLYN / POLYD

WPLDR(i) - W-Plane Transfer Function Load, Root Form
WROOTi = ROOTN / ROOTD, WPTFi = PSYNTH(WROOTi)

WPMPY(i,j,k) - W-Plane Transfer Function Multiply
WPTFi = WPTFj * WPTFk

WPPRN(i) - W-Plane Transfer Function Print
Print out WPTFi

WPRTS(i) - Find Roots Of W-Plane Transfer Function
WROOTi = Roots of WPTFi

WPSUB(i,j,k) - W-Plane Transfer Function Subtract
WPTFi = WPTFj - WPTFk

WSXFM(i) - W-to-S Root Transformation
Find "s plane equivalent" of roots of WPTFi

WZXFM(i,j) - W-to-Z Plane Transform
ZPTFi = Bilinear transform of WPTFj

<u>MISCELLANEOUS OPERATORS</u>

DTERM(i,j) - Determinant Of Matrix $\underline{M}(s)$ With Substitution Of Vector $\underline{B}(s)$ For Use In Transfer Function Evaluation Via Cramer's Method
POLYi = det $\underline{M}(s)$ with column j replaced with $\underline{B}(s)$

DETRM(i) - Old Version Of Operator DTERM
(No substitution of $\underline{B}(s)$)
POLYi = det $\underline{M}(s)$

STORE(i)   - Store Data From Current Batch Job
(printout suppressed if i.EQ.0)

RESTORE(i)   - Restore Data From Old Batch Job Or Interactive Session
(printout suppressed if i.EQ.0)

8.0 EXAMPLES

Examples are presented to demonstrate the utility of typical LCAP2 operators. Examples 1 through 12 were prepared to be executed sequentially in one batch job. Example 12 demonstrates the use of the STORE operator used to save transfer function, polynomial and matrix data for a restart capability. Example 13 demonstrates the RESTORE operator which restores data stored from a previous batch job.

Each of the examples begins with a statement of the problem and is followed by the user's FORTRAN code and the printed output of the program. To differentiate between FORTRAN code and comments, the FORTRAN code is in upper case and the comments are in lower case.

In Examples 1,5,6,7,9 and 10, which have requests for both printer and hardcopy plots, only the printer plots are shown. The hardcopy plot file generated by these examples is processed by the HARDCPY program after the LCAP2 program has been executed. The hardcopy plots from these examples are presented in Appendix F.

The examples in this section are essentially the same as those presented in the Interactive LCAP2 User's Guide, Ref. 3, so that differences in usage between both versions of LCAP2 can be compared. It should be noted that Batch LCAP2 can include FORTRAN expressions as part of the data entry, but was not used in these examples since the problems all used numeric data.

EXAMPLE 1 S-PLANE FREQUENCY RESPONSE

Problem: Compute frequency response of

$$\frac{25}{s^2 + 5s + 25}$$

between .10 to 100. rad/sec.

Data will be loaded in coefficient form using the operator SPLDC. Polynomial arrays POLYN and POLYD are used with SPLDC.

The FORTRAN code for this example is:

```
C     EXAMPLE 1
C     LOAD IN DATA USING LCAP2 OPERATOR SPLDC AND POLYNOMIAL ARRAYS
C     POLYN AND POLYD.
C
      POLYN(1)=0.           "deg. of num."
      POLYN(2)=25.          "coeff. of s**0"
      POLYD(1)=2.           "deg. of denom."
      POLYD(2)=25.          "coeff. of s**0"
      POLYD(3)=5.           "coeff. of s**1"
      POLYD(4)=1.           "coeff. of s**2"
      CALL SPLDC(1)         "load coefficient data into SPTF1"
C     . . . . . . . .
C     ENTER FREQUENCY RESPONSE PARAMETERS FOR USE WITH SFREQ
C
      FAUTO=1               ".NE.0 (preset=1) for auto. freq. selection mode"
      RAD=1                 ".NE.0 (preset=1) for rad/sec, otherwise hz"
      NOMEG=3               "number of values of OMEGA to be entered"
      OMEGA(1)=.1           "OMEGA(1)=first frequency value to be used"
      OMEGA(2)=1.           "user specified frequency value"
      OMEGA(3)=100.         "OMEGA(NOMEG)=last frequency value to be used"
      FDLAY=0               "time delay (preset=0)"
      FNICO=1               ".NE.0 (preset=0) for Nichols plot"
      FBODE=1               ".NE.0 (preset=1) for Bode plot"
      CYCLE=0               ".EQ.0 for auto. selection of 2 or 3 cycle for
                             Bode plots (1 cycle not available)"
      FNYQS=1               ".NE.0 (preset=0) for Nyquist plot"
      NQDB=1                ".NE.0 (preset=0) for hardcopy Nyquist plot in db"
      GRAFP=1               ".NE.0 (preset=1) for printer plot"
      FILM=1                ".NE.0 (preset=0) for hardcopy (high resolution)
                             plot"
```

```
C     ENTER PLOT TITLE
C     HEAD(1)-HEAD(7)   for 1ST LINE, PRINTER PLOT AND HARDCOPY
C     HEAD(8)-HEAD(14)  for 2ND LINE, HARDCOPY ONLY
C     HEAD(15)-HEAD(21) for 3RD LINE, HARDCOPY ONLY
C     HEAD(22)-HEAD(28) for 4TH LINE, HARDCOPY ONLY
C     FIRST ARGUMENT OF HEADIN4 IS POINTER TO ARRAY HEAD
      CALL HEADIN4(1,40HEXAMPLE 1 S PLANE FREQUENCY RESPONSE      )
      CALL SFREQ(1)        "compute frequency response of SPTF1"
C     . . . . . . . .
```

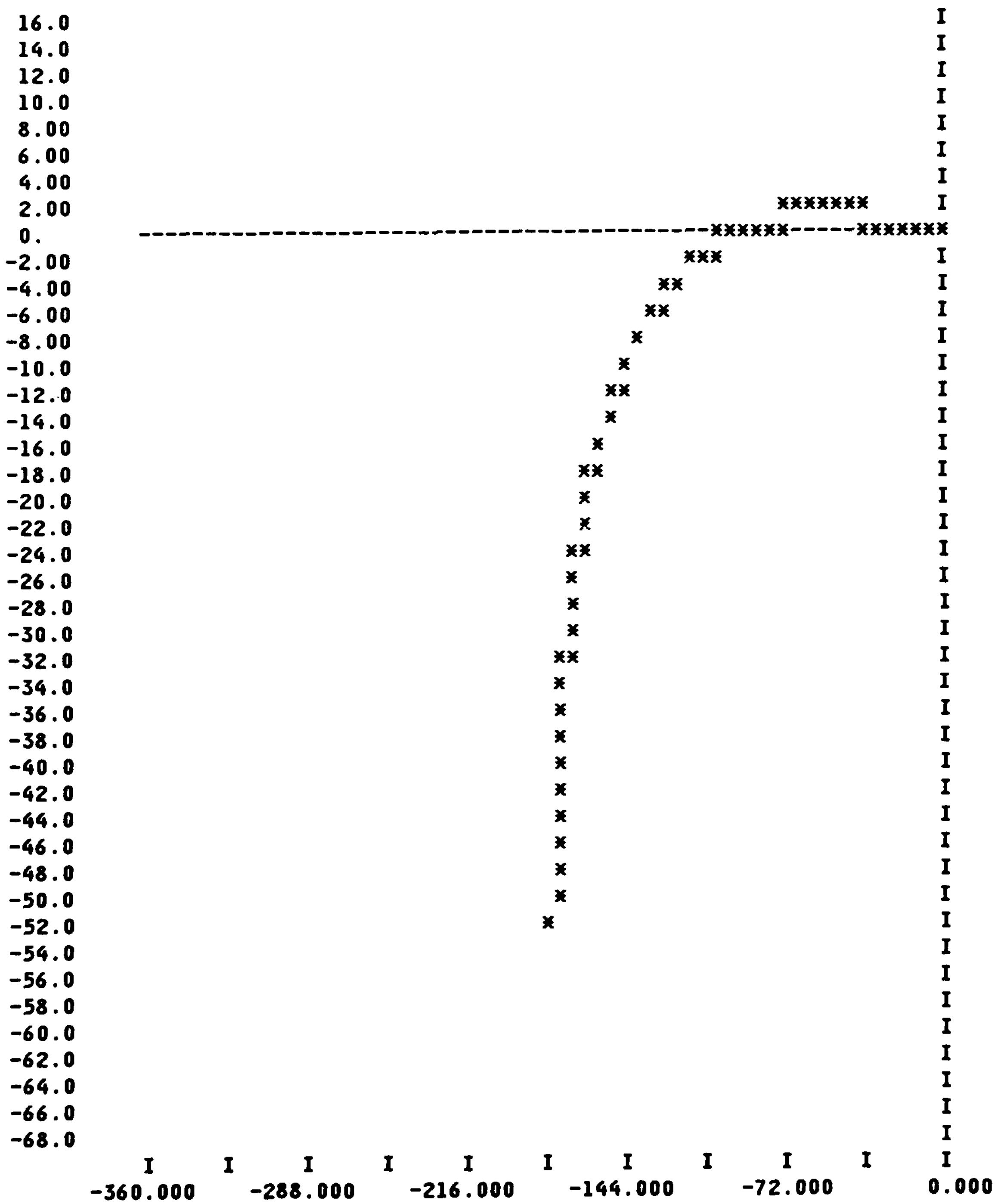
The printer output for this example is:

```
DEGREE OF POLYN  IS  0          (COEFFICIENTS IN ASCENDING ORDER)
25.

DEGREE OF POLYD  IS  2          (COEFFICIENTS IN ASCENDING ORDER)
25. 5. 1.

************************************************************
*    SPLDC - LOAD TRANSFER FUNCTION IN COEFFICIENT FORM    *
************************************************************

DEGREE OF NUMERATOR OF SPTF1  IS  0    (COEFFICIENTS IN ASCENDING ORDER)
25.

DEGREE OF DENOMINATOR OF SPTF1  IS  2  (COEFFICIENTS IN ASCENDING ORDER)
25. 5. 1.

BODE GAIN =    1.0000000

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                                                               CP=  4.97

************************************************************
*     SFREQ - FREQUENCY RESPONSE OF S-PLANE TRANSFER       *
*             FUNCTION  1                                  *
************************************************************

TRANSPORT DELAY OR DEAD TIME FOR S-PLANE FREQUENCY RESPONSE (FDLAY) =0.

AUTOMATIC FREQUENCY MODE IF FAUTO.NE.0, FAUTO = 1.000

NOMEG = 3.000    OMEGA = .1000     ,  1.000    , 100.     ,
```

OMEGA RAD/SEC	REAL	IMAGINARY	DB	PHASE	PHASE MARGIN
.1000	.100E+01	-.200E-01	.002	-1.15	178.85
.1200	.100E+01	-.240E-01	.003	-1.38	178.62
.1400	.100E+01	-.288E-01	.004	-1.65	178.35
.1728	.100E+01	-.346E-01	.005	-1.98	178.02
.2074	.100E+01	-.415E-01	.007	-2.38	178.62
.2488	.100E+01	-.499E-01	.011	-2.86	177.14
.2986	.100E+01	-.599E-01	.015	-3.43	176.57
.3583	.100E+01	-.720E-01	.022	-4.12	175.88
.4300	.100E+01	-.866E-01	.032	-4.95	175.05
.5160	.100E+01	-.104E+00	.046	-5.95	174.05
.6192	.100E+01	-.126E+00	.066	-7.17	172.83
.7430	.100E+01	-.152E+00	.095	-8.64	171.36
.8916	.999E+00	-.184E+00	.136	-10.44	169.56
1.070	.998E+00	-.224E+00	.194	-12.64	167.36
1.284	.995E+00	-.274E+00	.276	-15.37	164.63
2.084	.965E+00	-.487E+00	.673	-26.77	153.23
2.724	.889E+00	-.688E+00	1.017	-37.77	142.23
3.364	.728E+00	-.894E+00	1.236	-50.87	129.13
3.844	.539E+00	-.101E+01	1.202	-61.99	118.01
4.324	.311E+00	-.107E+01	.907	-73.74	106.26
4.804	.828E-01	-.103E+01	.320	-85.42	94.58
5.000	.225E-04	-.100E+01	.000	-90.00	90.00
5.480	-.162E+00	-.883E+00	-.940	-100.40	79.60
6.120	-.285E+00	-.701E+00	-2.421	-112.15	67.85
6.920	-.332E+00	-.503E+00	-4.399	-123.48	56.52
7.880	-.317E+00	-.336E+00	-6.707	-133.27	46.73
8.840	-.273E+00	-.231E+00	-8.834	-140.25	39.75
10.12	-.226E+00	-.148E+00	-11.362	-146.83	33.17
11.40	-.184E+00	-.999E-01	-13.584	-151.50	28.57
13.00	-.144E+00	-.651E-01	-16.014	-155.71	24.29
14.92	-.111E-00	-.418E-01	-18.536	-159.32	20.68
16.84	-.874E-01	-.285E-01	-20.731	-161.96	18.04
19.40	-.661E-01	-.183E-01	-23.275	-164.57	15.43
21.96	-.517E-01	-.124E-01	-25.488	-166.50	13.50
25.16	-.394E-01	-.816E-02	-27.902	-168.31	11.69
29.00	-.297E-01	-.528E-02	-30.410	-169.92	10.08
32.84	-.232E-01	-.361E-02	-32.598	-171.14	8.86
37.96	-.173E-01	-.232E-02	-35.140	-172.37	7.63
43.08	-.135E-01	-.158E-02	-37.354	-173.29	6.71
49.48	-.102E-01	-.104E-02	-39.774	-174.17	5.83
57.16	-.765E-02	-.674E-02	-42.292	-174.96	5.04
64.84	-.595E-02	-.461E-03	-44.489	-175.56	4.44
75.08	-.443E-02	-.297E-03	-47.043	-176.17	3.83
85.32	-.343E-02	-.202E-03	-49.268	-176.63	3.37
98.12	-.260E-02	-.133E-03	-51.700	-177.08	2.92
100.0	-.250E-02	-.125E-03	-52.030	-177.13	2.87

NICHOLS PLOT (MAGN. VS PHASE) 10/31/83

EXAMPLE 1 S PLANE FREQUENCY RESPONSE



FILM PLOT COMPLETED

```
BODE PLOT (MAGN. VS FREQ.)                                        10/31/83
EXAMPLE 1 S PLANE FREQUENCY RESPONSE

  16.0     I
  14.0     I
  12.0     I
  10.0     I
  6.00     I
  4.00     I
  2.00     I                         ****
  0.       **-**-**-***-**-**********----**----------------------------
 -2.00     I                               *
 -4.00     I                                *
 -6.00     I                                **
 -8.00     I                                 **
 -10.0     I                                  **
 -12.0     I                                   **
 -14.0     I                                    *
 -16.0     I                                     *
 -18.0     I                                      *
 -20.0     I                                      **
 -22.0     I                                       **
 -24.0     I                                        **
 -26.0     I                                         **
 -28.0     I                                          **
 -30.0     I                                           *
 -32.0     I                                            *
 -34.0     I                                             *
 -36.0     I                                             **
 -38.0     I                                              **
 -40.0     I                                               **
 -42.0     I                                                **
 -44.0     I                                                 **
 -46.0     I                                                  *
 -48.0     I                                                   **
 -50.0     I                                                    *
 -52.0     I                                                     *
 -54.0     I
 -56.0     I
 -58.0     I
 -60.0     I
 -62.0     I
 -64.0     I
 -66.0     I
 -68.0     I
           I     I     I     I     I     I     I     I     I     I     I
         .100E+00          1.00              10.0              100.
```

BODE PLOT (MAGN. VS FREQ.)   10/31/83
EXAMPLE 1 S PLANE FREQUENCY RESPONSE

```
  30.0     I
  20.0     I
  10.0     I
 0.        XX-XX-XX-XXX------------------------------------------------
 -10.0     I            XX XXXX
 -20.0     I                   XXXX
 -30.0     I                       XX
 -40.0     I                        XXX
 -50.0     I                          XX
 -60.0     I                           XX
 -70.0     I                            XX
 -80.0     I                             X
 -90.0     I                             XX
 -100.     I                              XX
 -110.     I                               X
 -120.     I                                X
 -130.     I                                XX
 -140.     I                                 XXX
 -150.     I                                   XXX
 -160.     I                                     XXXX
 -170.     I                                        XXXXXXXXXX
 -180.     I                                                 XXXXX
 -190.     I
 -200.     I
 -210.     I
 -220.     I
 -230.     I
 -240.     I
 -250.     I
 -260.     I
 -270.     I
 -280.     I
 -290.     I
 -300.     I
 -310.     I
 -320.     I
 -330.     I
 -340.     I
 -350.     I
 -360.     I
 -370.     I
 -380.     I
 -390.     I
           I     I     I     I     I     I     I     I     I     I     I
         .100E+00         1.00              10.0              100.
       FILM PLOT COMPLETED
```

NYQUIST PLOT 10/31/83

EXAMPLE 1 S PLANE FREQUENCY RESPONSE

```
    4.20                                  I
    4.00                                  I
    3.80                                  I
    3.60                                  I
    3.40                                  I
    3.20                                  I
    3.00                                  I
    2.80                                  I
    2.60                                  I
    2.40                                  I
    2.20                                  I
    2.00                                  I
    1.80                                  I
    1.60                                  I
    1.40                                  I
    1.20                                  I
    1.00                                  I
    .800                                  I
    .600                                  I
    .400                                  I
    .200                                  I
0.          ----------------------------***-----------*------------------
   -.200                              *** I           *
   -.400                              *   I           *
   -.600                              *   I          *
   -.800                               ** I        ***
   -1.00                                 ***********
   -1.20                                  I
   -1.40                                  I
   -1.80                                  I
   -2.00                                  I
   -2.20                                  I
   -2.40                                  I
   -2.60                                  I
   -2.80                                  I
   -3.00                                  I
   -3.20                                  I
   -3.40                                  I
   -3.80                                  I
   -4.00                                  I
   -4.20                                  I
            I     I     I     I     I     I     I     I     I     I     I
        -2.50 -2.00 -1.50 -1.00  -.50  0.00   .50  1.00  1.50  2.00  2.50
        FILM PLOT COMPLETED                                  CP=  5.30
```

EXAMPLE 2 ROOTS OF A POLYNOMIAL

Problem: Compute the roots of the polynomial

$$x^3 + 13x^2 + 38x^1 + 34$$

The FORTRAN code for this example is:

```
C     EXAMPLE 2
C     LOAD IN DATA USING LCAP2 OPERATOR PRTS AND POLYNOMIAL ARRAY POLY
C
      POLY(1)=3              "deg. of polynomial"
      POLY(2)=34.            "coeff. of x**0"
      POLY(2)=38.            "coeff. of x**1"
      POLY(2)=13.            "coeff. of x**2"
      POLY(2)=1.             "coeff. of x**3"
      CALL PLDC(1)           "load coefficient data into POLY1"
C     . . . . . . .
      CALL PRTS(1)           "find roots of POLY1"
C     . . . . . . .
```

The printer output for this case is:

```
DEGREE OF POLY   IS  3          (COEFFICIENTS IN ASCENDING ORDER)
34. 38. 13. 1.

************************************************************
*     PLDC - POLYNOMIAL LOAD IN COEFFICIENT FORM           *
************************************************************

DEGREE OF POLY1  IS  3          (COEFFICIENTS IN ASCENDING ORDER)
34. 38. 13. 1.

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                                                   CP=  5.30

DEGREE OF POLY1  IS  3          (COEFFICIENTS IN ASCENDING ORDER)
34. 38. 13. 1.

************************************************************
*     PRTS - FIND ROOTS OF POLY1                           *
************************************************************
```

THE ROOTS OF ROOT1 ARE

NO.	REAL	IMAG.	OMEGA	ZETA
1	-1.8445105	-.49938380	1.9109168	.96524896
2	-1.8445105	.49938380	1.9109168	.96524896
3	-9.3109790	0.		

LOW ORDER NON ZERO COEFFICIENT = 34.000000

DEGREE OF POLY IS 3 (COEFFICIENTS IN ASCENDING ORDER)
34. 38. 13. 1.

- -

CP= 5.31

EXAMPLE 3 MULTIPLY TWO S-PLANE TRANSFER FUNCTIONS

Problem: Multiply the following two s-plane transfer functions

$$\frac{25}{s^2 + 5s + 25} \quad \text{and} \quad \frac{30\left(\frac{s}{10} + 1\right)}{70(s)\left(\frac{s}{7} + 1\right)\left(\frac{s}{1 + j2} + 1\right)\left(\frac{s}{1 - j2} + 1\right)}$$

The first transfer function is the same as SPTF1 from Example 1. The second transfer function is given in root form. When transfer function data is expressed in root form, gains associated with the numerator and denominator must be specified to uniquely define the transfer function. The gains used by LCAP2 correspond to the low order non-zero coefficient of the numerator and denominator polynomials if the root form were expanded out. For this example, these two gains would be 30 and 70. (A distinction is made for the low order non-zero coefficient since the low order coefficient can be zero as is that of the denominator in this example.)

The FORTRAN code for this example is:

```
C      EXAMPLE 3
C      FIRST TRANSFER FUNCTION IS THE SAME AS SPTF1 OF EXAMPLE 1
C
C      LOAD IN ROOT DATA FOR THE SECOND TRANSFER FUNCTION USING
C      SUBROUTINE SPLDR AND POLYNOMIAL ROOT ARRAYS ROOTN AND ROOTD
C
       ROOTN(1)=(1.,30.)          "real part = no. of roots and imag.
                                   part = low non-zero gain of num.,"
       ROOTN(2)=(-10.,0.)         "first root"
       ROOTD(1)=(4.,70.)          "real part = no. of roots and imag.
                                   part = low non-zero gain of denom.,"
       ROOTD(2)=(-1.,2.)          "first root"
       ROOTD(3)=(-1.,-2.)         "second root"
       ROOTD(4)=(-7.,0.)          "third root"
       ROOTD(5)=(0.,0.)           "fourth root"
       CALL SPLDR(2)              "load root data into SPTF2"
C      . . . . . . .
C      MULTIPLY THE TWO TRANSFER FUNCTIONS
       CALL SPMPY(3,1,2)          "multiply SPTF1 and SPTF2 and store
                                   in SPTF3"
C      . . . . . . .
```

The printer output for this example is:

THE ROOTS OF ROOTN ARE

NO.	REAL	IMAG.	OMEGA	ZETA
1	-10.000000	0.		

THE ROOTS OF ROOTD ARE

NO.	REAL	IMAG.	OMEGA	ZETA
1	-1.0000000	2.0000000		
2	-1.0000000	-2.0000000		
3	-7.0000000	0.		
4	0.	0.		

```
***********************************************************
*     SPLDR - LOAD TRANSFER FUNCTION IN ROOT FORM         *
***********************************************************
```

THE NUMERATOR ROOTS OF SROOT2 ARE

NO.	REAL	IMAG.	OMEGA	ZETA
1	-10.000000	0.		

LOW ORDER NON ZERO COEFFICIENT = 30.000000

THE DENOMINATOR ROOTS OF SROOT2 ARE

NO.	REAL	IMAG.	OMEGA	ZETA
1	-1.0000000	2.0000000	2.23606798	.447213595
2	-1.0000000	-2.0000000	2.23606798	.447213595
3	-7.0000000	0.		
4	0.	0.		

LOW ORDER NON ZERO COEFFICIENT = 70.000000

DEGREE OF NUMERATOR OF SPTF2 IS 1 (COEFFICIENTS IN ASCENDING ORDER)
30. 3.
DEGREE OF DENOMINATOR OF SPTF2 IS 4 (COEFFICIENTS IN ASCENDING ORDER)
0. 70. 38. 18. 2.

BODE GAIN = .42857143

CP= 5.31

```
DEGREE OF NUMERATOR OF SPTF1  IS  0    (COEFFICIENTS IN ASCENDING ORDER)
25.

DEGREE OF DENOMINATOR OF SPTF1  IS  2  (COEFFICIENTS IN ASCENDING ORDER)
25. 5. 1.

BODE GAIN =     1.000000


                    THE NUMERATOR ROOTS OF SROOT2  ARE

NO.          REAL            IMAG.             OMEGA            ZETA

 1      -10.000000       0.

                    LOW ORDER NON ZERO COEFFICIENT =    30.000000

                    THE DENOMINATOR ROOTS OF SROOT2  ARE

NO.          REAL            IMAG.             OMEGA            ZETA

 1      -1.0000000       2.0000000
 2      -1.0000000      -2.0000000
 3      -7.0000000       0.

                    LOW ORDER NON ZERO COEFFICIENT =    70.000000

DEGREE OF NUMERATOR OF SPTF2  IS  1    (COEFFICIENTS IN ASCENDING ORDER)
30. 3.
DEGREE OF DENOMINATOR OF SPTF2  IS  4  (COEFFICIENTS IN ASCENDING ORDER)
0. 70. 38. 18. 2.

BODE GAIN =     .42857143

***********************************************
*   SPTF3  =     SPTF1      *    SPTF2        *
***********************************************

DEGREE OF NUMERATOR OF SPTF3  IS  1    (COEFFICIENTS IN ASCENDING ORDER)
750. 75.

DEGREE OF DENOMINATOR OF SPTF3  IS  6  (COEFFICIENTS IN ASCENDING ORDER)
0. 1750. 1300. 710. 178. 28. 2.

BODE GAIN =     .42857143

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                                                          CP=  5.32
```

Note that if the second transfer function were expressed in a different form, the numerator and denominator non-zero coefficients would not necessarily appear explicitly. For example, if the root locus representation were used, i.e.,

$$\frac{3(s + 10)}{2(s + 7)(s + 1 + j2)(s + 1 - j2)}$$

then:

numerator low order non-zero coefficient = (3)(10) = 30
denominator low order non-zero coefficient = (2)(7)(1+j2)(1-j2) = 70

EXAMPLE 4 CLOSED LOOP S-PLANE TRANSFER FUNCTION

Problem: Given open loop transfer function G(s) and feedback transfer H(s) where

$$G(s) = \frac{15}{s^3 + 6s^2 + 5s^1}$$

$$H(s) = \frac{s + 1.5}{s + 15}$$

compute the closed loop transfer function F(s) where

$$F(s) = \frac{G(s)}{1 + G(s)H(s)}$$

The FORTRAN code for this example is:

```
C     EXAMPLE 4
C     LOAD IN COEFFICIENTS OF G(S)
C
      POLYN(1)=0.          "deg. of num."
      POLYN(2)=15.         "coeff. of s**0"
      POLYD(1)=3.          "deg. of denom."
      POLYD(2)=0.          "coeff. of s**0"
      POLYD(3)=5.          "coeff. of s**1"
      POLYD(4)=6.          "coeff. of s**2"
      POLYD(5)=1.          "coeff. of s**3"
      CALL REMARK3(30HG(S) IS IN SPTF4
      CALL SPLDC(4)        "load G(s) data into SPTF4"
C     . . . . . . . .
C     LOAD IN COEFFICIENTS OF H(S)
C
      POLYN(1)=1.          "deg. of num."
      POLYN(2)=1.5         "coeff. of s**0"
      POLYN(3)=1.          "coeff. of s**1"
      POLYD(1)=1.          "deg. of denom."
      POLYD(2)=15.         "coeff. of s**0"
      POLYD(3)=1.          "coeff. of s**1"
      CALL REMARK3(30HH(S) IS IN SPTF5
```

```
      CALL SPLDC(5)         "load H(s) data into SPTF5"
C     . . . . . . . .
C     MULTIPLY G(S) AND H(S)
C
      CALL REMARK4(40HG(S)*H(S) IS IN SPTF6                   )
      CALL SPMPY(6,4,5)    "G(s)*H(s) is in SPTF6"
C     . . . . . . . .
C     LOAD IN UNITY TRANSFER FUNCTION
C
      POLYN(1)=0           "deg. of num."
      POLYN(2)=1           "coeff. of s**0"
      POLYD(1)=0           "deg. of denom."
      POLYD(2)=1           "coeff. of s**0"
      CALL SPLDC(7)        "unity transfer function is in SPTF7"
C     . . . . . . . .
C
      CALL REMARK4(40H1 + G(S)*G(S) IS IN SPTF8               )
      CALL SPADD(8,7,6)    "1 + G(S)*H(S) is in SPTF8"
C     . . . . . . . .
C     FORM CLOSED LOOP TRANSFER FUNCTION BY DIVIDING G(S) BY G(S)+H(S)
C
      CALL SPDIV(9,4,8)    "closed loop transfer function is in SPTF9"
C     . . . . . . . .
C     ELIMINATE COMMON ROOTS BETWEEN NUMERATOR AND DENOMINATOR
C
      CALL REMARK4(40HCLOSED LOOP T. F. IS IN SPTF9           )
      CALL SELCR(9)        "eliminate common roots of SPTF4"
C     . . . . . . . .
```

The printer output for this example is:

```
G(S) IS IN SPTF4

DEGREE OF POLYN  IS  0          (COEFFICIENTS IN ASCENDING ORDER)
15.

DEGREE OF POLYD  IS  2          (COEFFICIENTS IN ASCENDING ORDER)
0. 5. 6. 1.

************************************************************
*     SPLDC - LOAD TRANSFER FUNCTION IN COEFFICIENT FORM    *
************************************************************

DEGREE OF NUMERATOR OF SPTF4  IS  0    (COEFFICIENTS IN ASCENDING ORDER)
15.
```

```
DEGREE OF DENOMINATOR OF SPTF4  IS  3   (COEFFICIENTS IN ASCENDING ORDER)
0. 5. 6. 1.

BODE GAIN =    3.0000000

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                                                 CP=  5.33
H(S) IS IN SPTF5

DEGREE OF POLYN  IS  1           (COEFFICIENTS IN ASCENDING ORDER)
1.5 1.

DEGREE OF POLYD  IS  1           (COEFFICIENTS IN ASCENDING ORDER)
0. 1.

*************************************************************
*     SPLDC - LOAD TRANSFER FUNCTION IN COEFFICIENT FORM    *
*************************************************************

DEGREE OF NUMERATOR OF SPTF5  IS  1     (COEFFICIENTS IN ASCENDING ORDER)
1.5 1.

DEGREE OF DENOMINATOR OF SPTF5  IS  1   (COEFFICIENTS IN ASCENDING ORDER)
15. 1.

BODE GAIN =    .10000000E+00

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                                                            CP=  5.33

G(S)*H(S) IS IN SPTF6

DEGREE OF NUMERATOR OF SPTF4  IS  0     (COEFFICIENTS IN ASCENDING ORDER)
15.

DEGREE OF DENOMINATOR OF SPTF4  IS  3   (COEFFICIENTS IN ASCENDING ORDER)
0. 5. 6. 1.

BODE GAIN =    3.0000000

DEGREE OF NUMERATOR OF SPTF5  IS  1     (COEFFICIENTS IN ASCENDING ORDER)
1.5 1.

DEGREE OF DENOMINATOR OF SPTF5  IS  1   (COEFFICIENTS IN ASCENDING ORDER)
15. 1.

BODE GAIN =    .10000000E+00
```

```
*********************************************
*   SPTF6  =      SPTF4       *    SPTF5          *
*********************************************

DEGREE OF NUMERATOR OF SPTF6  IS  1     (COEFFICIENTS IN ASCENDING ORDER)
22.5 15.

DEGREE OF DENOMINATOR OF SPTF6  IS  4   (COEFFICIENTS IN ASCENDING ORDER)
0. 75. 95. 21. 1.

BODE GAIN =     .30000000

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                                                            CP=  5.35

DEGREE OF POLYN  IS  0           (COEFFICIENTS IN ASCENDING ORDER)
1.

DEGREE OF POLYD  IS  0           (COEFFICIENTS IN ASCENDING ORDER)
1.

*********************************************************
*     SPLDC - LOAD TRANSFER FUNCTION IN COEFFICIENT FORM     *
*********************************************************

DEGREE OF NUMERATOR OF SPTF7  IS  0     (COEFFICIENTS IN ASCENDING ORDER)
1.

DEGREE OF DENOMINATOR OF SPTF7  IS  0   (COEFFICIENTS IN ASCENDING ORDER)
1.

BODE GAIN =    1.00000000

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                                                            CP=  5.36

1 + G(S)*H(S) IS IN SPTF8

DEGREE OF NUMERATOR OF SPTF7  IS  0     (COEFFICIENTS IN ASCENDING ORDER)
1.

DEGREE OF DENOMINATOR OF SPTF7  IS  0   (COEFFICIENTS IN ASCENDING ORDER)
1.

BODE GAIN =    1.00000000

DEGREE OF NUMERATOR OF SPTF6  IS  1     (COEFFICIENTS IN ASCENDING ORDER)
22.5 15.
```

```
DEGREE OF DENOMINATOR OF SPTF6  IS  4   (COEFFICIENTS IN ASCENDING ORDER)
0. 75. 95. 21. 1.

BODE GAIN =    .30000000

************************************************
*   SPTF8  =     SPTF7      +    SPTF6         *
************************************************

DEGREE OF NUMERATOR OF SPTF8  IS  4     (COEFFICIENTS IN ASCENDING ORDER)
22.5 90. 95. 21. 1.

DEGREE OF DENOMINATOR OF SPTF8  IS  4   (COEFFICIENTS IN ASCENDING ORDER)
0. 75. 95. 21. 1.0000000

BODE GAIN =    .30000000

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                                                                  CP=  5.38

DEGREE OF NUMERATOR OF SPTF4  IS  0     (COEFFICIENTS IN ASCENDING ORDER)
15.

DEGREE OF DENOMINATOR OF SPTF4  IS  3   (COEFFICIENTS IN ASCENDING ORDER)
0. 5. 6. 1.

BODE GAIN =    3.0000000

DEGREE OF NUMERATOR OF SPTF8  IS  4     (COEFFICIENTS IN ASCENDING ORDER)
22.5 90. 95. 21. 1.

DEGREE OF DENOMINATOR OF SPTF8  IS  4   (COEFFICIENTS IN ASCENDING ORDER)
0. 75. 95. 21. 1.0000000

BODE GAIN =    .30000000

*************************************************
*   SPTF9  =     SPTF4      /    SPTF8          *
*************************************************

DEGREE OF NUMERATOR OF SPTF9  IS  4      (COEFFICIENTS IN ASCENDING ORDER)
0. 1125. 1425. 315. 15.

DEGREE OF DENOMINATOR OF SPTF9  IS  7     (COEFFICIENTS IN ASCENDING ORDER)
0. 112.5 585. 1037.5 765. 226. 27. 1.

BODE GAIN =    10.000000
```

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                                                                  CP=  5.40

CLOSED LOOP T.F. IS IN SPTF9

DEGREE OF NUMERATOR OF SPTF9  IS  4       (COEFFICIENTS IN ASCENDING ORDER)
0. 1125. 1425. 315. 15.

DEGREE OF DENOMINATOR OF SPTF9  IS  7     (COEFFICIENTS IN ASCENDING ORDER)
0. 112.5 585. 1037.5 765. 226. 27. 1.

BODE GAIN =     10.3.0000000

************************************************************
*     SELCR - ELIMINATE COMMON ROOTS OF S-PLANE            *
*             TRANSFER FUNCTION  9                         *
************************************************************

                     THE COMMON ROOTS ELIMINATED ARE

NO.        REAL             IMAG.              OMEGA            ZETA

 1      -1.0000000       0.
 2      -5.0000000       0.
 3      0.               0.

**********************************************
*   SPTF9  =PSYNTH(SROOT9 )                  *
**********************************************

                    THE NUMERATOR ROOTS OF SROOT9  ARE

NO.        REAL             IMAG.              OMEGA            ZETA

 1      -15.000000       0.

                    LOW ORDER NON ZERO COEFFICIENT =    1125.0000

                    THE DENOMINATOR ROOTS OF SROOT9  ARE

NO.        REAL             IMAG.              OMEGA            ZETA

 1      -.755541434     0.
 2      -.416857608     0.
 3      -4.73265645     0.
 4      -15.0949445     0.

                    LOW ORDER NON ZERO COEFFICIENT =    112.50000
```

DEGREE OF NUMERATOR OF SPTF9 IS 1   (COEFFICIENTS IN ASCENDING ORDER)
1125. 75.

DEGREE OF DENOMINATOR OF SPTF9 IS 4 (COEFFICIENTS IN ASCENDING ORDER)
112.5 450. 475. 105. 5.

BODE GAIN =   10.000000

CP= 5.44

EXAMPLE 5 S-PLANE ROOT LOCUS

Problem: Compute the root locus for the following transfer function

$$\frac{1.8\left(\frac{s}{1}+1\right)}{s\left(\frac{s}{2}+1\right)\left(\frac{s}{3}+1\right)\left(\frac{s}{5-j1}+1\right)\left(\frac{s}{5+j1}+1\right)}$$

by varying the nominal gain from .125 to 2.

The FORTRAN code for this example is:

```
C     EXAMPLE 5
C     LOAD IN ROOT DATA FOR TRANSFER FUNCTION
C
      ROOTN(1)=(1.,1.8)     "real part = no. of roots and imag.
                             part = low non-zero gain of num.,"
      ROOTN(2)=(-1.,0.)     "first root"
      ROOTD(1)=(5.,1.)      "real part = no. of roots and imag.
                             part = low non-zero gain of denom.,"
      ROOTD(2)=(-5.,-1.)    "first root"
      ROOTD(3)=(-5.,1.)     "second root"
      ROOTD(4)=(-3.,0.)     "third root"
      ROOTD(5)=(-2.,0.)     "fourth root"
      ROOTD(6)=(0.,0.)      "fifth root"
      CALL SPLDR(10)        "data stored in SPTF10"
C     . . . . . . . .
C     ENTER ROOT LOCUS PARAMETERS FOR USE WITH SLOCI
C
      NLOCI=2               "number of values of KGAIN to be entered
      KGAIN(1)=.125         "KGAIN(1)=first gain value to be used
      KGAIN(2)=2.           "KGAIN(NLOCI)=last gain value to be used
      KFLG=0                ".EQ.0 to increment gain by multiplying by KDELT,
                             .NE.0 to increment gain by adding by KDELT"
      KDELT=2               "value for changing gains (preset=1.E4 so that no
                             additional gains are computed by the program"
      GRAFP=1               ".NE.0 (preset=1) for printer plot
      FILM=1                ".NE.0 (preset=0) for hardcopy (high resolution)
                             plot"
      RLXMN=-9              "min. x axis for plot"
      RLXMX=1               "max. x axis for plot"
                            "auto. scaling of x axis if RLXMN=RLXMX"
      RLYMN=-1              "min. y axis for plot"
      RLYMX=9               "max. y axis for plot"
                            "auto. scaling of y axis if RLYMN=RLYMX"
```

```
      CALL HEADIN4(1,40HEXAMPLE 5 S PLANE ROOT LOCUS            )
      CALL SLOCI(10)       "compute root loci of SPTF10"
```

The printer output for this example is:

```
                                THE ROOTS OF ROOTN  ARE

NO.        REAL             IMAG.              OMEGA            ZETA

 1      -1.0000000       0.

                                THE ROOTS OF ROOTD  ARE

NO.        REAL             IMAG.              OMEGA            ZETA

 1      -5.0000000      -1.0000000
 2      -5.0000000      -1.0000000
 3      -3.0000000      0.
 4      -2.0000000      0.
 5     0.               0.

************************************************************
*    SPLDR = LOAD TRANSFER FUNCTION IN ROOT FORM           *
************************************************************

                         THE NUMERATOR ROOTS OF SROOT10  ARE

NO.        REAL             IMAG.              OMEGA            ZETA

 1      -1.0000000       0.

                         LOW ORDER NON ZERO COEFFICIENT =    1.8000000

                         THE DENOMINATOR ROOTS OF SROOT10 ARE

NO.        REAL             IMAG.              OMEGA            ZETA

 1      -5.0000000      -1.0000000
 2      -5.0000000      -1.0000000
 3      -3.0000000      0.
 4      -2.0000000      0.
 5     0.               0.

                         LOW ORDER NON ZERO COEFFICIENT =    1.0000000

DEGREE OF NUMERATOR OF SPTF10 IS  1    (COEFFICIENTS IN ASCENDING ORDER)
1.8 1.8
```

DEGREE OF DENOMINATOR OF SPTF10 IS 5 (COEFFICIENTS IN ASCENDING ORDER)
0. 1. 12.17948717949 .525641025641 .09615384615385 .006410256410256

BODE GAIN = 1.8000000

CP= 5.47

THE NUMERATOR ROOTS OF SROOT10 ARE

NO.	REAL	IMAG.	OMEGA	ZETA
1	-1.0000000	0.		

LOW ORDER NON ZERO COEFFICIENT = 1.8000000

THE DENOMINATOR ROOTS OF SROOT10 ARE

NO.	REAL	IMAG.	OMEGA	ZETA
1	-5.0000000	-1.0000000		
2	-5.0000000	-1.0000000		
3	-3.0000000	0.		
4	-2.0000000	0.		
5	0.	0.		

LOW ORDER NON ZERO COEFFICIENT = 1.0000000

DEGREE OF NUMERATOR OF SPTF10 IS 1 (COEFFICIENTS IN ASCENDING ORDER)
1.8 1.8

DEGREE OF DENOMINATOR OF SPTF10 IS 5 (COEFFICIENTS IN ASCENDING ORDER)
0. 1. 12.17948717949 .525641025641 .09615384615385 .006410256410256

BODE GAIN = 1.8000000

```
********************************************************
*     SLOCI - ROOT LOCUS OF S-PLANE TRANSFER           *
*             FUNCTION 10                              *
********************************************************
```

THE OPEN LOOP ZEROES ARE

NO.	REAL	IMAG.	OMEGA	ZETA
1	-1.000000	0.		

THE OPEN LOOP POLES ARE

NO.	REAL	IMAG.	OMEGA	ZETA
1	-5.0000000	-1.0000000		
2	-5.0000000	1.0000000		
3	-2.0000000	0.		
4	-3.0000000	0.		
5	0.	0.		

CLOSED LOOP POLES FOR GAIN = .1250000 (GAIN NO. 1) ARE

NO.	REAL	IMAG.	OMEGA	ZETA
1	-1.7947341	-1.1077310	2.1090611	.85096357
2	-1.7947341	1.1077310	2.1090611	.85096357
3	-5.5892217	1.6614254	5.8309290	.95854737
4	-5.5892217	-1.6614254	5.8309290	.95854737
5	-.23208832	0.		

CLOSED LOOP POLES FOR GAIN = .2500000 (GAIN NO. 2) ARE

NO.	REAL	IMAG.	OMEGA	ZETA
1	-1.4171985	-1.4418858	2.0217532	.70097500
2	-1.4171985	1.4418858	2.0217532	.70097500
3	-5.8577544	1.9606112	6.1771582	.94829277
4	-5.8577544	-1.9606112	6.1771582	.94829277
5	-.45009415	0.		

CLOSED LOOP POLES FOR GAIN = .5000000 (GAIN NO. 3) ARE

NO.	REAL	IMAG.	OMEGA	ZETA
1	-.94735061	-1.9000842	2.1231564	.44619915
2	-.94735061	1.9000842	2.1231564	.44619915
3	-6.1975818	-2.3343777	6.6226384	.93581763
4	-6.1975818	2.3343777	6.6226384	.93581763
5	-.71013518	0.		

CLOSED LOOP POLES FOR GAIN = 1.0000000 (GAIN NO. 4) ARE

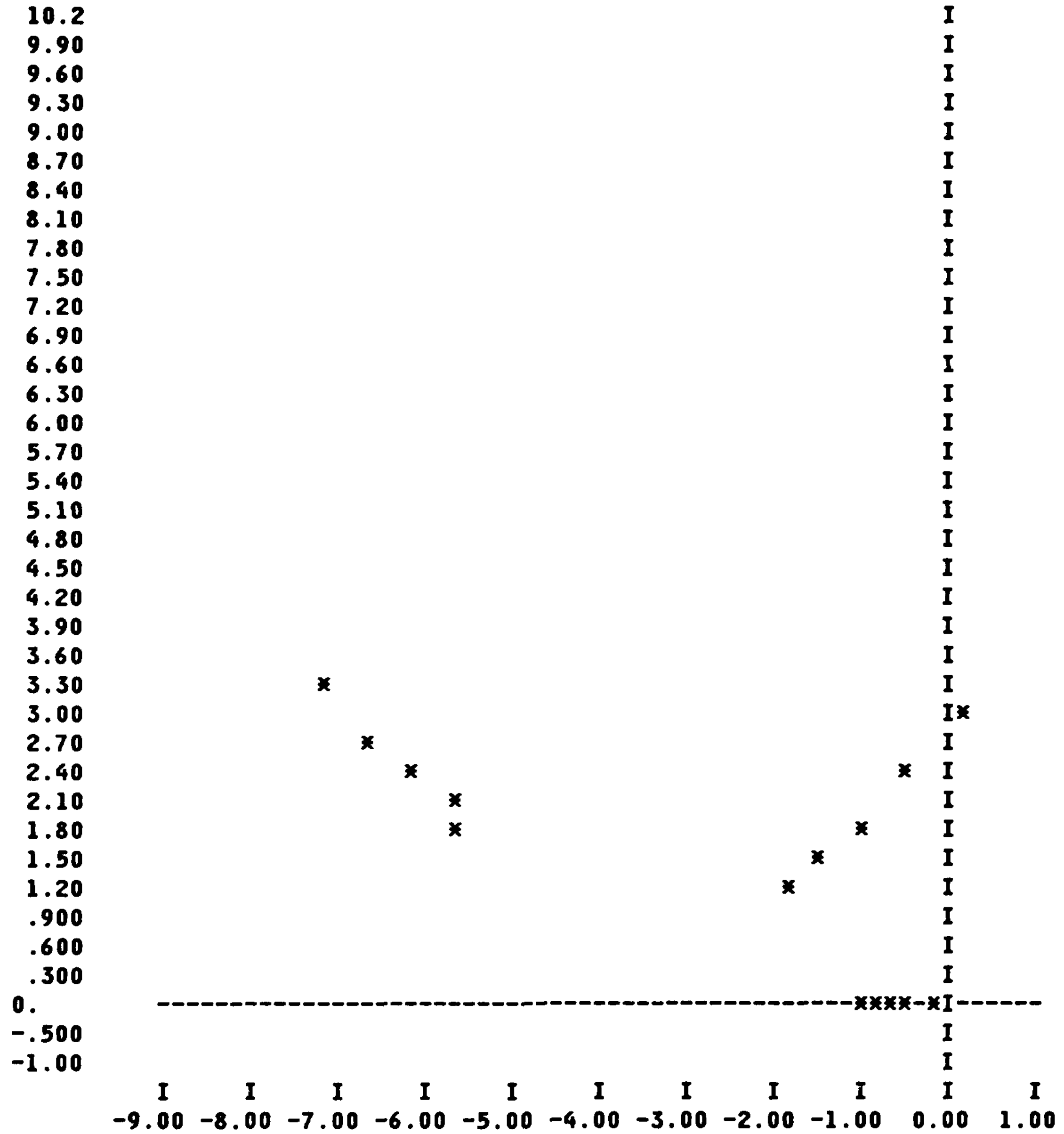
NO.	REAL	IMAG.	OMEGA	ZETA
1	-.44922549	-2.4680974	2.5086468	.17907084
2	-.44922549	2.4680974	2.5086468	.17907084
3	-6.6183282	-2.7904455	7.1825382	.92144699
4	-6.6183282	2.7904455	7.1825382	.92144699
5	-.86489263	0.		

CLOSED LOOP POLES FOR GAIN = 2.0000000 (GAIN NO. 5) ARE

NO.	REAL	IMAG.	OMEGA	ZETA
1	.10025104	-3.1088677	3.1104837	-.32230049E-01
2	.10025104	3.1088677	3.1104837	-.32230049E-01
3	-7.1323100	-3.3395771	7.8754442	.90563907
4	-7.1323100	3.3395771	7.8754442	.90563907
5	-.93588209	0.		

FILM PLOT COMPLETED

ROOT LOCUS 10/31/83
EXAMPLE 5 S PLANE ROOT LOCUS



CP= 5.56

EXAMPLE 6 INVERSE LAPLACE TRANSFORM AND TIME RESPONSE

Problem: Find the inverse Laplace transform and the step response of the following transfer function

$$\frac{432s + 4320}{s^4 + 35s^3 + 345s^2 + 1008s^1 + 2160}$$

Plot the response in increments of .05 seconds from 0 to 5 seconds.

The inverse Laplace transform is computed by the partial fraction method. The algorithm used to compute the partial fraction expansion requires that (1) there are no multiple poles other than those at the origin and (2) the degree of the numerator must not be greater than the number of non-zero poles of the denominator. Up to 5 poles at the origin are allowed.

The FORTRAN code for this example is:

```
C     EXAMPLE 6
C     LOAD COEFFICIENT DATA FOR TRANSFER FUNCTION
C
      POLYN(1)=1.          "deg. of num."
      POLYN(2)=4320.       "coeff. of s**0"
      POLYN(3)=432.        "coeff. of s**1"
      POLYD(1)=4.          "deg. of denom."
      POLYD(2)=2160.       "coeff. of s**0"
      POLYD(3)=1008.       "coeff. of s**1"
      POLYD(4)=345.        "coeff. of s**2"
      POLYD(5)=35.         "coeff. of s**3"
      POLYD(5)=1.          "coeff. of s**4"
      CALL SPLDC(11)       "load coefficient data into SPTF11"
C     . . . . . . . .
C     ENTER PARAMETERS FOR TIME RESPONSE
C
      TSTEP=1              ".NE.0 (preset=1) for step response,
                            .EQ.0 for impulse response"
      TMAGN=1.             "magnitude of the input"
      TZERO=0.             "starting time for evaluating the response"
      TEND=5.              "end time for evaluating the response"
      TDELT=.05            "time increment for evaluating the response"
      GRAFP=1              ".NE.0 (preset=1) for printer plot
      FILM=1               ".NE.0 (preset=0) for higher resolution electro-
                            static plot"
      CALL HEADIN5(1,
     +50HEXAMPLE 6 INVERSE LAPLACE TRANSFORM AND TIME RESPO
      CALL HEADIN1(6,10HNSE       )
```

```
      CALL STIME(11)       "compute inverse Laplace transform and time response"
C     . . . . . . .
```

The printer output for this example is:

```
DEGREE OF POLYN  IS  1 (COEFFICIENTS IN ASCENDING ORDER)
4320. 432.

DEGREE OF POLYD  IS  4           (COEFFICIENTS IN ASCENDING ORDER)
2160. 1008. 345. 35. 1.

*************************************************************
*     SPLDC - LOAD TRANSFER FUNCTION IN COEFFICIENT FORM    *
*************************************************************

DEGREE OF NUMERATOR OF SPTF11 IS  1     (COEFFICIENTS IN ASCENDING ORDER)
4320. 432.

DEGREE OF DENOMINATOR OF SPTF11 IS  4   (COEFFICIENTS IN ASCENDING ORDER)
2160. 1008. 345. 35. 1.

BODE GAIN =    2.0000000

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                                                          CP=  5.58

                    THE NUMERATOR ROOTS OF SROOT11 ARE

NO.        REAL             IMAG.              OMEGA            ZETA

 1      -10.000000       0.

                    LOW ORDER NON ZERO COEFFICIENT =    4320.0000

                    THE DENOMINATOR ROOTS OF SROOT11 ARE

NO.        REAL             IMAG.              OMEGA            ZETA

 1      -1.5000000      -2.5980762        3.00000000       .500000000
 2      -1.5000000       2.5980762        3.00000000       .500000000
 3      -12.000000      0.
 4      -20.000000      0.

                    LOW ORDER NON ZERO COEFFICIENT =    2160.0000

DEGREE OF NUMERATOR OF SPTF11 IS  1    (COEFFICIENTS IN ASCENDING ORDER)
4320. 432.
```

```
DEGREE OF DENOMINATOR OF SPTF11 IS  4  (COEFFICIENTS IN ASCENDING ORDER)
2160. 1008. 345. 35. 1.

BODE GAIN =    2.0000000


************************************************************
*     STIME - TIME RESPONSE OF S-PLANE TRANSFER            *
*             FUNCTION 11                                  *
************************************************************

COMPUTE STEP RESPONSE IF TSTEP.NE.0., (TSTEP =  1.0 )
OTHERWISE COMPUTE IMPULSE RESPONSE.

SCALE OUTPUT BY TMAGN, (TMAGN =   1.00000      )

 NO.                ROOT                     PARTIAL FRACTION COEFFICIENT

 1     -1.5000000       -2.5980762       -.99977959        -.69735056
 2     -1.5000000        2.5980762       -.99977959         .69735056
 3     -12.000000       0.                 .76923077E-01   0.
 4     -20.000000       0.               -.76923077E-01   0.
 5     0.               0.                20.000000       0.

ANALYTICAL SOLUTION IS THE SUMMATION OF THE FOLLOWING  4 TERMS


                                                     (  2.0000     ) * T** 0
((-2.000    )*(cos( 2.60     *T))+(-1.395     )*(sin( 2.60     *T)))*E**(-1.500    *T)
                                             (  .76923E-01) * E**( -12.000     * T)
                                             ( -.77364E-01) * E**( -20.000     * T)
```

*** TIME RESPONSE ***

TIME	VALUE	TIME	VALUE	TIME	VALUE
0.	0.	1.7000	2.1497	3.4000	2.0054
.50000E-01	.66945E-02	1.7500	2.1235	3.4500	2.0066
.10000	.44037E-01	1.8000	2.0985	3.5000	2.0075
.15000	.10879	1.8500	2.0749	3.5500	2.0082
.20000	.20673	1.9000	2.0530	3.6000	2.0086
.25000	.32913	1.9500	2.0330	3.6500	2.0087
.30000	.46993	2.0000	2.0152	3.7000	2.0086
.35000	.62345	2.0500	1.9995	3.7500	2.0084
.40000	.78455	2.1000	1.9860	3.8000	2.0084
.45000	.94871	2.1500	1.9747	3.8500	2.0076
.50000	1.1120	2.2000	1.9655	3.9000	2.0070
.55000	1.2711	2.2500	1.9582	3.9500	2.0063
.60000	1.4233	2.3000	1.9529	4.0000	2.0057
.65000	1.5664	2.3500	1.9493	4.0500	2.0050
.70000	1.6985	2.4000	1.9472	4.1000	2.0042
.75000	1.8186	2.4500	1.9466	4.1500	2.0035
.80000	1.9257	2.5000	1.9471	4.2000	2.0029
.85000	2.0194	2.5500	1.9487	4.2500	2.0022
.90000	2.0997	2.6000	1.9511	4.3000	2.0016
.95000	2.1667	2.6500	1.9542	4.3500	2.0011
1.0000	2.2209	2.7000	1.9578	4.4000	2.0006
1.0500	2.2209	2.7500	1.9618	4.4500	2.0001
1.1000	2.2937	2.8000	1.9661	4.5000	1.9998
1.1500	2.3140	2.8500	1.9704	4.5500	1.9994
1.2000	2.3249	2.9000	1.9748	4.6000	1.9992
1.2500	2.3276	2.9500	1.9791	4.6500	1.9990
1.3000	2.3230	3.0000	1.9832	4.7000	1.9988
1.3500	2.3123	3.0500	1.9871	4.7500	1.9987
1.4000	2.2966	3.1000	1.9907	4.8000	1.9986
1.4500	2.2769	3.1500	1.9940	4.8500	1.9986
1.5000	2.2542	3.2000	1.9970	4.9000	1.9986
1.5500	2.2293	3.2500	1.9996	4.9500	1.9986
1.6000	2.2032	3.3000	2.0019	5.0000	1.9987
1.6500	2.1764	3.3500	2.0038		

TIME RESPONSE 10/31/83
EXAMPLE 6 INVERSE LAPLACE TRANSFORM AND TIME RESPONSE

```
 3.30     I
 3.20     I
 3.10     I
 3.00     I
 2.90     I
 2.80     I
 2.70     I
 2.60     I
 2.50     I
 2.40     I
 2.30     I            ******
 2.20     I          **      **
 2.10     I          *         ****
 2.00     I         *             ******  *****************************
 1.90     I         *                  ****
 1.80     I        *
 1.70     I       *
 1.60     I       *
 1.50     I
 1.40     I      *
 1.30     I      *
 1.20     I
 1.10     I     *
 1.00     I
 .900     I    *
 .800     I    *
 .700     I
 .600     I   *
 .500     I   *
 .400     I
 .300     I  *
 .200     I *
 .100E+00 I *
0.        **-----------------------------------------------------------
-.100E+00 I
-.200     I
-.300     I
-.400     I
-.500     I
-.600     I
-.700     I
-.800     I
-.900     I
          I     I     I     I     I     I     I     I     I     I     I
        0.00   .50  1.00  1.50  2.00  2.50  3.00  3.50  4.00  4.50  5.00
       FILM PLOT COMPLETED
```

EXAMPLE 7 INVERSE Z-TRANSFORM BY POWER SERIES METHOD

Problem: Find the inverse z transform and the step response of the following transfer function

$$\frac{a_4 z^4 + a_3 z^3 + a_2 z^2 + a_1 z^1 + a_0}{b_4 z^4 + b_3 z^3 + b_2 z^2 + b_1 z^1 + b_0}$$

where $a_0 = -1.96474786E-4$ $\quad b_0 = 0.243334776$

$a_1 = -3.87975526E-4$ $\quad b_1 = -1.58617798$

$a_2 = 1.49221356E-5$ $\quad b_2 = 3.43416283$

$a_3 = 4.07871707E-4$ $\quad b_3 = -3.09127983$

$a_4 = 2.01448831E-4$ $\quad b_4 = 1.0$

and the sampling period is .04 seconds.

Although the power series method for computing the inverse z transform is not as accurate as the partial fraction method, the results for typical transfer functions are very good. To provide a measure of the accuracy of the response, the results are computed in both single and double precision and compared.

The FORTRAN code for this example is:

```
C     EXAMPLE 7
C     LOAD IN COEFFICIENT DATA INTO ZPTF1
C
      POLYN(1)=4.                "deg. of num."
      POLYN(2)=-1.96474786E-4    "coeff. of z**0"
      POLYN(3)=-3.87975526E-4    "coeff. of z**1"
      POLYN(4)=1.49221356E-5     "coeff. of z**2"
      POLYN(5)=4.07871707E-4     "coeff. of z**3"
      POLYN(6)=2.01448831E-4     "coeff. of z**4"
```

```
      POLYD(1)=4.                "deg. of denom."
      POLYD(2)=.243334776        "coeff. of z**0"
      POLYD(3)=-1.58617798       "coeff. of z**1"
      POLYD(4)=3.43416283        "coeff. of z**2"
      POLYD(5)=-3.09127983       "coeff. of z**3"
      POLYD(6)=1.                "coeff. of z**4"
      CALL ZPLDC(1)              "load transfer function into ZPTF11"
C     . . . . . . . .
C     ENTER TIME RESPONSE PARAMETERS FOR USE WITH ZTIME
C
      TSTEP=1                    ".NE.0 (preset=1) for step response,
                                  .EQ.0 for impulse response"
      TMAGN=1                    "magnitude of the input"
      TEND=2.0                   "end time for evaluating the response"
      SAMPT=.04                  "sampling period"
      CALL HEADIN5(1,
     +50HEXAMPLE 7 INVERSE Z TRANSFROM AND TIME RESPONSE           )
      CALL HEADIN1(6,10H         )
      CALL ZTIME(1)              "compute inverse z transform of ZPTF1"
C     . . . . . . . .
```

The printer output for this example is:

```
DEGREE OF POLYN  IS  1            (COEFFICIENTS IN ASCENDING ORDER)
-.000196474786 -.000387975526 .0000149221356 .000407871707
.000201448831

DEGREE OF POLYD  IS  4            (COEFFICIENTS IN ASCENDING ORDER)
.243334776 -1.58617798 3.43416283 -3.09127983 1.

**************************************************************
*     ZPLDC - LOAD TRANSFER FUNCTION IN COEFFICIENT FORM     *
**************************************************************

DEGREE OF NUMERATOR OF ZPTF1  IS  1     (COEFFICIENTS IN ASCENDING ORDER)
-.000196474786 -.000387975526 .0000149221356 .000407871707
.000201448831

DEGREE OF DENOMINATOR OF ZPTF1  IS  4   (COEFFICIENTS IN ASCENDING ORDER)
.243334776 -1.58617798 3.43416283 -3.09127983 1.

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                                                          CP=  5.71

DEGREE OF NUMERATOR OF ZPTF1  IS  1     (COEFFICIENTS IN ASCENDING ORDER)
-.000196474786 -.000387975526 .0000149221356 .000407871707
.000201448831
```

```
DEGREE OF DENOMINATOR OF ZPTF1  IS  4   (COEFFICIENTS IN ASCENDING ORDER)
.243334776 -1.58617798 3.43416283 -3.09127983 1.


***************************************************************
*     ZTIME - TIME RESPONSE OF Z-PLANE TRANSFER               *
*             FUNCTION  1                                     *
***************************************************************

COMPUTE STEP RESPONSE IF TSTEP.NE.0., (TSTEP =  1.0 )
OTHERWISE COMPUTE IMPULSE RESPONSE.

SCALE OUTPUT BY TMAGN, (TMAGN =  1.00000      )

FINAL VALUE =   .999909
```

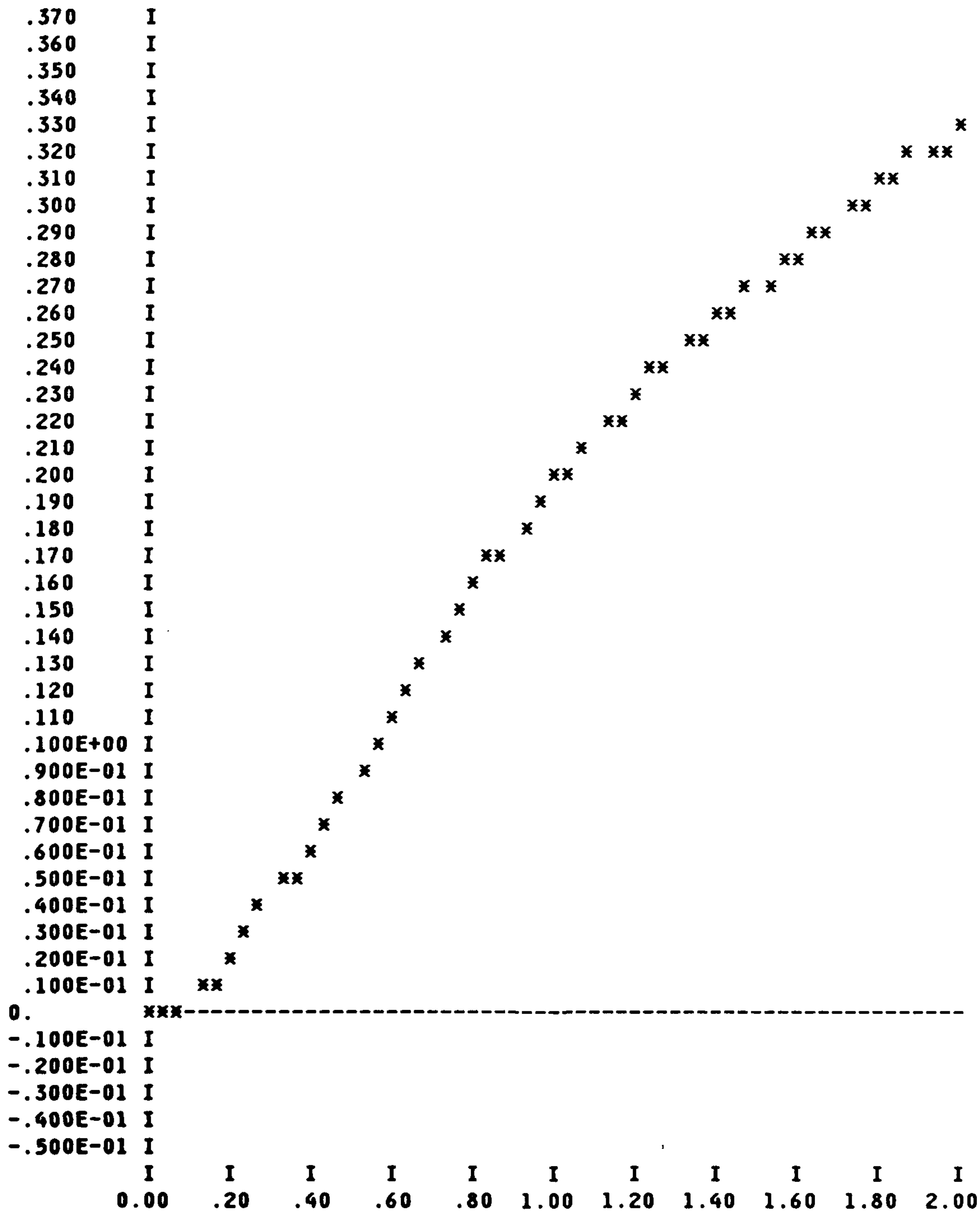
*** TIME RESPONSE ***

TIME	VALUE	TIME	VALUE	TIME	VALUE
0.	.20135E-03	.68000	.13117	1.3600	.25310
.40000E-01	.12321E-02	.72000	.14013	1.4000	.25856
.80000E-01	.37411E-02	.76000	.14886	1.4400	.26388
.12000	.78894E-02	.80000	.15736	1.4800	.26907
.16000	.13486E-01	.84000	.16562	1.5200	.27413
.20000	.20269E-01	.88000	.17364	1.5600	.27907
.24000	.27989E-01	.92000	.18142	1.6000	.28391
.28000	.36424E-01	.96000	.18897	1.6400	.28864
.32000	.45388E-01	1.0000	.19629	1.6800	.22927
.36000	.54723E-01	1.0400	.20339	1.7200	.29781
.40000	.64299E-01	1.0800	.21027	1.7600	.30227
.44000	.74009E-01	1.1200	.21695	1.8000	.30664
.48000	.83764E-01	1.1600	.22342	1.8400	.31093
.52000	.93493E-01	1.2000	.22970	1.8800	.31516
.56000	.10314	1.2400	.23581	1.9200	.31931
.60000	.11266	1.2800	.24173	1.9600	.32340
.64000	.12201	1.3200	.24750	2.0000	.32743

AT THE LAST TIME POINT THE SINGLE PRECISION VALUE DIFFERS FROM THE
DOUBLE PRECISION VALUE BY   .16E-08 PERCENT.

TIME RESPONSE 10/31/83

EXAMPLE 7 INVERSE Z TRANSFORM AND TIME RESPONSE



FILM PLOT COMPLETED

EXAMPLE 8 Z TRANSFORM OF AN S-PLANE TRANSFER FUNCTION

Problem: Compute the z transform of the following transfer function with a zero order hold and a delay of .008 seconds. The sampling period is .08 seconds.

$$\frac{2900s + 2900}{s^5 + 4s^4 + 124s^3 + 363s^2}$$

The partial fraction expansion method is used to compute the z transform. The algorithm used to compute the partial fraction expansion requires that (1) there are no multiple poles other than those at the origin and (2) the degree of the numerator must not be greater than the number of non-zero poles of the denominator. Up to 5 poles at the origin are allowed (this includes the one from the zero order hold if applicable).

The FORTRAN code for this example is:

```
C     EXAMPLE 8
C     LOAD COEFFICIENT DATA INTO SPTF12
C
      POLYN(1)=1.           "deg. of num."
      POLYN(2)=2900.        "coeff. of s**0"
      POLYN(3)=2900.        "coeff. of s**1"
      POLYD(1)=5.           "deg. of denom."
      POLYD(2)=0.           "coeff. of s**0"
      POLYD(3)=0.           "coeff. of s**1"
      POLYD(4)=363.         "coeff. of s**2"
      POLYD(5)=124.         "coeff. of s**3"
      POLYD(6)=4.           "coeff. of s**4"
      POLYD(7)=1.           "coeff. of s**5"
      CALL SPLDC(12)        "load G(s) data into SPTF12"
C     . . . . . . .
C     ENTER PARAMETERS FOR USE WITH SZXFM
C
      SAMPT=.08             "sampling period"
      DELAY=.008            "time delay (preset=0),
                             (enter a negative value for time advance)"
      ZOH=1                 ".NE.0 (preset=1) for inclusion of zero order hold"
      CALL SZXFM(2,12)      "compute z transform of SPTF12 and store
                             into ZPTF2"
C     . . . . . . .
```

The printer output for this example is:

```
------------------------------------------------------------------------------
                                                                 CP=   5.76

DEGREE OF POLYN  IS  1            (COEFFICIENTS IN ASCENDING ORDER)
2900. 2900.

DEGREE OF POLYD  IS  4            (COEFFICIENTS IN ASCENDING ORDER)
0. 0. 363. 124. 4. 1.

*************************************************************
*     SPLDC - LOAD TRANSFER FUNCTION IN COEFFICIENT FORM    *
*************************************************************

DEGREE OF NUMERATOR OF SPTF12 IS  1     (COEFFICIENTS IN ASCENDING ORDER)
2900. 2900.

DEGREE OF DENOMINATOR OF SPTF12 IS  4   (COEFFICIENTS IN ASCENDING ORDER)
0. 0. 363. 124. 4. 1.

BODE GAIN =     7.9889807

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                                                                 CP=   5.78

                    THE NUMERATOR ROOTS OF SROOT12 ARE

NO.          REAL              IMAG.              OMEGA             ZETA

 1      -1.0000000        0.

                    LOW ORDER NON ZERO COEFFICIENT =    2900.0000

                    THE DENOMINATOR ROOTS OF SROOT12 ARE

NO.          REAL              IMAG.              OMEGA             ZETA

 1      -.50000000      -10.988630        11.0000000       .45454545E-01
 2      -.50000000       10.988630        11.0000000       .45454545E-01
 3      -3.0000000       0.
 4      0.               0.
 5      0.               0.

                    LOW ORDER NON ZERO COEFFICIENT =    363.00000

DEGREE OF NUMERATOR OF SPTF12 IS  1    (COEFFICIENTS IN ASCENDING ORDER)
2900. 2900.
DEGREE OF DENOMINATOR OF SPTF12 IS  5  (COEFFICIENTS IN ASCENDING ORDER)
0. 0. 363. 124. 4. 1.

BODE GAIN =     7.9889807
```

```
****************************************
*  ZROOT2  =    SZXFM OF      SROOT12  *
****************************************

****************************************
*   ZPTF2  =    SZXFM OF      SPTF12   *
****************************************
```

SAMPLING PERIOD, SAMPT = .0800 DELAY = .0080 ZOH = 1

NO.	ROOT		PARTIAL FRACTION EXPANSION COEFFICIENT	
1	-.50000000	-10.988630	.96684606E-01	-.40456452E-02
2	-.50000000	10.988630	-.96684606E-01	.40456452E-02
3	-3.0000000	0.	1.6914552	0.
4	0.		-1.8848244	0.
5	0.	0.	5.2599625	0.
6	0.	0.	7.9889807	0.

THE NUMERATOR ROOTS OF ZROOT2 ARE

NO.	REAL	IMAG.	OMEGA	ZETA
1	-12.268933	0.		
2	-.67135130E-04	0.		
3	.92311634	0.		
4	-.13469859	0.		
5	-1.1567519	0.		

LOW ORDER NON ZERO COEFFICIENT = -.23106189E-04

THE DENOMINATOR ROOTS OF ZROOT2 ARE

NO.	REAL	IMAG.	OMEGA	ZETA
1	.61284138	-.73996066	.96078944	-.63785191
2	.61284138	.73996006	.96078944	-.63785191
3	.78662786	0.		
4	1.00000000	0.		
5	1.00000000	0.		
6	0.	0.		

LOW ORDER NON ZERO COEFFICIENT = -46.473538

DEGREE OF NUMERATOR OF ZPTF2 IS 5 (COEFFICIENTS IN ASCENDING ORDER)
-.00002310618894132 -.3443427037346 -2.507846683653 .6794170629569
2.464719122914 .1950347342772

```
DEGREE OF DENOMINATOR OF ZPTF2  IS  6  (COEFFICIENTS IN ASCENDING ORDER)
0. -46.47353837271 213.7325205421 -416.8323061287 442.3612041218
-256.7878801625 64.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                                                            CP=  5.84
```

EXAMPLE 9 MULTIRATE FREQUENCY RESPONSE BY FREQUENCY DECOMPOSITION

Problem: Compute the frequency response of the following function

$$\frac{1}{n}\sum_{k=1}^{n-1} G_n^{T/n}\left(z_n e^{j2*pi*k/n}\right)$$

where T=.24 seconds, n=3 and $G^{T/n}(z_n)$ is the z-transform computed in Example 9.

The above function is Sklansky's frequency decomposition method for expressing the output transform of a fast to slow sampler in terms of the faster input transfer function. An example of how this frequency decomposition method can be applied to the stability analysis of a multiloop multirate control system is given in Example 2 of Ref. 4. The above function can represent the open loop transfer function of Eq (6.6) in Ref. 4.

The operator ZMRFQ evaluates the frequency response of the above function by using only the transform of the faster sampled signal $G^{T/n}(z_n)$. The response is computed by the indicated summation with shifted values of z_n. This operation yields only the multirate frequency response of $G^{T/n}(z_n)$. No z-transform at the slower sampling rate is computed. In the next example though, an explicit form for the implementing the frequency decomposition method is described.

Since one-half of the sampling frequency of the slower output sampler is 13.09 rad/sec, the frequency range to be used for this example will be 1.0 to 13.0 rad/sec.

The FORTRAN code for this example is:

```
C      EXAMPLE 9
C      TRANSFER FUNCTION IS ZPTF2 FROM PREVIOUS CASE
C
C      ENTER FREQUENCY RESPONSE PARAMETERS FOR ZMRFQ
C
       NOMEG=2                "number of values of OMEGA to be entered"
       OMEGA(1)=1.            "OMEGA(1)=first frequency value to be used"
       OMEGA(2)=13.           "OMEGA(NOMEG)=last frequency value to be used"
       SAMPT=.24              "sampling period of slower output sampler"
```

```
      CALL REMARK5(
     +50HZPTF2 IS AT THE FASTER SAMPLING RATE, SAMPT=.08    )
C     INTEGER RATIO OF OUTPUT/INPUT SAMPLING PERIODS IS ENTERED AS THE
C     SECOND ARGUMENT OF ZMRFQ(I,M)
C
      CALL REMARK5(
     +50HFREQ. RESP. WILL BE AT THE SLOWER SAMPLING RATE,   )
      CALL REMARK2(20HSAMPT=.08           )
      CALL HEADIN5(1,
     +50HEXAMPLE 9 MULTIRATE FREQUENCY RESPONSE BY FREQUENC)
      CALL HEADIN2(6,20HY DECOMPOSITION     )
      CALL ZMRFQ(2,3)       "compute multirate frequency response of ZPTF2"
C     . . . . . . .
```

The printer output for this example is:

ZPTF2 IS AT THE FASTER SAMPLING RATE, SAMPT=.08

FREQ. RESP. WILL BE AT THE SLOWER SAMPLING RATE,

SAMPT=.08

THE NUMERATOR ROOTS OF ZROOT2 ARE

NO.	REAL	IMAG.	OMEGA	ZETA
1	-12.268933	0.		
2	-.67135130E-04	0.		
3	.92311634	0.		
4	-.13469859	0.		
5	-1.1567519	0.		

LOW ORDER NON ZERO COEFFICIENT = -.23106189E-04

THE DENOMINATOR ROOTS OF ZROOT2 ARE

NO.	REAL	IMAG.	OMEGA	ZETA
1	.61284138	-.73996066	.96078944	-.63785191
2	.61284138	.73996006	.96078944	-.63785191
3	.78662786	0.		
4	1.00000000	0.		
5	1.00000000	0.		
6	0.	0.		

LOW ORDER NON ZERO COEFFICIENT = -46.473538

DEGREE OF NUMERATOR OF ZPTF2 IS 5 (COEFFICIENTS IN ASCENDING ORDER)
-.00002310618894132 -.3443427037346 -2.507846683653 .6794170629569
2.464719122914 .1950347342772

DEGREE OF DENOMINATOR OF ZPTF2 IS 6 (COEFFICIENTS IN ASCENDING ORDER)
0. -46.47353837271 213.7325205421 -416.8323061287 442.3612041218
-256.7878801625 64.

```
****************************************************
*     ZMRFQ - MULTIRATE FREQUENCY RESPONSE (BY      *
*             FREQUENCY DECOMPOSITION) OF ZPTF2     *
****************************************************
```

SAMPLING PERIOD (SAMPT) = .2400

INTEGER RATIO OF OUTPUT/INPUT SAMPLING PERIODS, MTGER = 3

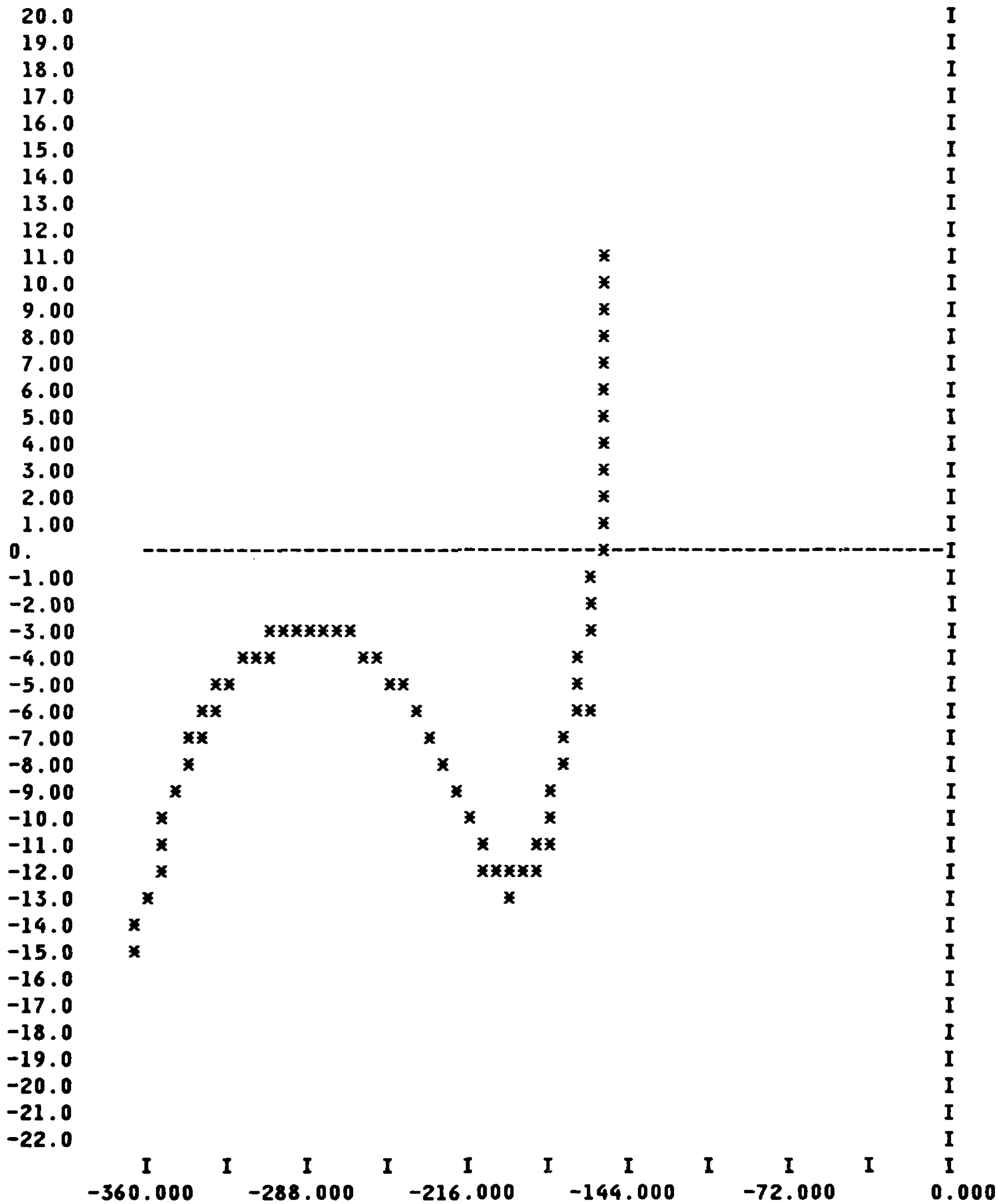
AUTOMATIC FREQUENCY MODE IF FAUTO.NE.0, FAUTO = 1.000

NOMEG = 2.000   OMEGA = 1.000   , 13.00   ,

OMEGA RAD/SEC	ZREAL	ZIMAG	REAL	IMAGINARY	DB	PHASE	PHASE MARGIN
1.000	.997	.080	-.331E+01	-.143E+01	11.125	-156.66	23.34
1.200	.995	.096	-.247E+01	-.113E+01	8.664	-155.48	24.52
1.400	.994	.112	-.195E+01	-.906E+00	6.648	-155.08	24.92
1.700	.991	.136	-.147E+01	-.670E+00	4.171	-155.51	24.49
2.000	.987	.159	-.117E+01	-.504E+00	2.125	-156.75	23.25
2.369	.982	.188	-.933E+00	-.360E+00	-.000	-158.91	21.09
2.869	.974	.220	-.722E+00	-.230E+00	-2.411	-162.36	17.64
3.469	.962	.274	-.560E+00	-.132E+00	-4.805	-166.75	13.25
4.269	.942	.335	-.423E+00	-.556E-01	-7.377	-172.53	7.47
5.069	.919	.394	-.339E+00	-.112E-01	-9.387	-178.11	1.89
5.346	.910	.415	-.317E+00	.458E-07	-9.966	-180.00	-.00
6.946	.850	.528	-.241E+00	.467E-01	-12.206	-190.98	-10.98
8.546	.775	.632	-.225E+00	.989E-01	-12.202	-203.76	-23.76
9.546	.722	.692	-.250E+00	.181E+00	-10.213	-215.89	-35.39
10.05	.694	.720	-.270E+00	.286E+00	-8.096	-226.62	-46.62
10.35	.677	.736	-.262E+00	.409E+00	-6.287	-237.50	-57.50
10.55	.655	.747	-.208E+00	.530E+00	-4.893	-248.60	-68.60
10.70	.656	.755	-.110E+00	.630E+00	-3.887	-260.08	-80.08
10.82	.648	.762	.250E-01	.688E+00	-3.247	-272.08	-92.08
10.92	.642	.767	.160E+00	.690E+00	-2.994	-283.03	-103.03
11.02	.636	.772	.292E+00	.640E+00	-3.054	-294.53	-114.53
11.12	.630	.777	.393E+00	.546E+00	-3.442	-305.70	-125.70
11.22	.623	.782	.447E+00	.435E+00	-4.097	-315.81	-135.81
11.37	.614	.789	.458E+00	.283E+00	-5.376	-328.22	-148.22
11.57	.601	.799	.409E+00	.150E+00	-7.222	-339.91	-159.91

11.82	.585	.811	.335E+00	.654E-01	-9.326	-348.97	-168.97
12.12	.566	.825	.270E+00	.236E-01	-11.343	-355.00	-175.00
12.62	.532	.847	.212E+00	.339E-02	-13.460	-359.09	-179.09
13.00	.506	.862	.198E+00	.333E-03	-14.077	-359.90	-179.90

NICHOLS PLOT (MAGN. VS PHASE) 10/31/83
EXAMPLE 9 MULTIRATE FREQUENCY RESPONSE BY FREQUENCY DECOMPOSITON



FILM PLOT COMPLETED

The z-plane frequencies in the tabulation of the response are at the faster sampling rate. Note that at 13.0 rad/sec the z-plane frequency is (.506 + j.852) or approximately 60 degrees on the unit circle. At the slower sampling rate this point would be at approximately the (-1. + j 0.) point, which is 3 times 60 degrees on the unit circle.

EXAMPLE 10 RATIONAL REPRESENTATION OF FREQUENCY DECOMPOSITION METHOD

Problem: Compute the frequency response of the function

$$\frac{1}{n}\sum_{k=1}^{n-1} G^{T/n}\left(z_n e^{j2*pi*k/n}\right)$$

using the explicit form of Sklansky's frequency decomposition method where T=.24 seconds, n=3 and $G^{T/n}(z_n)$ is the z-transform computed in Example 9.

In Example 9 the frequency response was computed using the operator ZMRFQ which numerically evaluated the function over a range of frequencies for z_n. Until recently, this was the only means in LCAP2 of applying the frequency decomposition method. A new operator, ZMRXFM, has been implemented which will yield a rational representation of this frequency decomposition method. This operator computes the output transform of a fast to slow sampler. This output transform is a transfer function represented in rational form at the slower sampling rate. After this transfer function has been computed, the single rate z-plane frequency response operator, ZFREQ, can be used to evaluate the response.

The numerical technique for computing the rational form of the frequency decomposition method will be documented in the near future.

The FORTRAN code for this example is:

```
C     EXAMPLE 10
C     G(Z) IS ZPTF2 FROM THE PREVIOUS EXAMPLE
C
C     ENTER PARAMETERS FOR ZMRXFM
C
      SAMPT=.24              "Sampling period of slower output sampler"
      NTGER=3                "integer ratio of output/input sampling periods"
      CALL REMARK5(
     +50ZPTF2 IS AT THE FASTER SAMPLING RATE, SAMPT=.08    )
      CALL REMARK5(
     +50HZPTF3 WILL BE AT THE SLOWER SAMPLING RATE,        ,)
      CALL REMARK2(20HSAMPT=.24           )
```

```
      CALL ZMRXFM(3,2)        "compute multirate transform of ZPTF2 and store
                               in ZPTF3"
C     . . . . . . . .
C     ZPTF3 IS AT THE SLOWER SAMPLING RATE
C
C     ENTER FREQUENCY RESPONSE PARAMETERS FOR USE WITH ZFREQ
C
      NOMEG=2                 "number of values of OMEGA to be entered"
      OMEGA(1)=1.             "OMEGA(1)=first frequency value to be used"
      OMEGA(2)=13.            "OMEGA(NOMEG)=last frequency value to be used"
      CALL HEADIN5(1,
     +50HEXAMPLE 10 RATIONAL REPRESENTATION OF FREQUENCY DE)
      CALL HEADIN2(6,20HCOMPOSITION METHOD  )
      CALL ZFREQ(3)           "compute frequency response of ZPTF3"
```

The printer output for this example is:

ZPTF2 IS AT THE FASTER SAMPLING RATE, SAMPT=.08

ZPTF3 WILL BE AT THE SLOWER SAMPLING RATE,

SAMPT=.24

THE NUMERATOR ROOTS OF ZROOT2 ARE

NO.	REAL	IMAG.	OMEGA	ZETA
1	-12.268933	0.		
2	-.67135130E-04	0.		
3	.92311634	0.		
4	-.13469859	0.		
5	-1.1567519	0.		

LOW ORDER NON ZERO COEFFICIENT = -.23106189E-04

THE DENOMINATOR ROOTS OF ZROOT2 ARE

NO.	REAL	IMAG.	OMEGA	ZETA
1	.61284138	-.73996066	.96078944	-.63785191
2	.61284138	.73996006	.96078944	-.63785191
3	.78662786	0.		
4	1.00000000	0.		
5	1.00000000	0.		
6	0.	0.		

LOW ORDER NON ZERO COEFFICIENT = -46.473538

```
DEGREE OF NUMERATOR OF ZPTF2  IS  5   (COEFFICIENTS IN ASCENDING ORDER)
-.00002310618894132 -.3443427037346 -2.507846683653 .6794170629569
2.464719122914 .1950347342772

DEGREE OF DENOMINATOR OF ZPTF2  IS  6  (COEFFICIENTS IN ASCENDING ORDER)
0. -46.47353837271 213.7325205421 -416.8323061287 442.3612041218
-256.7878801625 64.

***********************************************
*  ZROOT3  =    ZMRXFM OF      ZROOT2         *
***********************************************

***********************************************
*   ZPTF3  =PSYNTH(ZROOT3 )                   *
***********************************************


                    THE NUMERATOR ROOTS OF ZROOT3 ARE

NO.          REAL              IMAG.             OMEGA            ZETA

 1      -.10647750E-01    0.
 2      -.59161620        0.
 3       .78658382        0.
 4      -2.9655282        0.

                    LOW ORDER NON ZERO COEFFICIENT =    -.18162533

                    THE DENOMINATOR ROOTS OF ZROOT3 ARE

NO.          REAL              IMAG.             OMEGA            ZETA

 1      -.77650116        -.42857182        .88692044        .87550262
 2      -.77650116         .42857182        .88692044        .87550262
 3       .48675226        0.
 4      1.00000000        0.
 5      1.00000000        0.

                    LOW ORDER NON ZERO COEFFICIENT =    -24.505145

DEGREE OF NUMERATOR OF ZPTF3  IS  4    (COEFFICIENTS IN ASCENDING ORDER)
-.1816253320627 -17.19496629374 -12.53387821671 34.37676332934
12.3603697263

DEGREE OF DENOMINATOR OF ZPTF3  IS  5  (COEFFICIENTS IN ASCENDING ORDER)
-24.5051447024 50.97511989617 39.80519865903 -70.51517819695
-59.75999565584 64.

------------------------------------------------------------------------------
                                                                CP=  5.98
```

```
************************************************************
*     ZFREQ - FREQUENCY RESPONSE OF Z-PLANE TRANSFER       *
*               FUNCTION  3                                *
************************************************************
```

SAMPLING PERIOD (SAMPT) = .2400

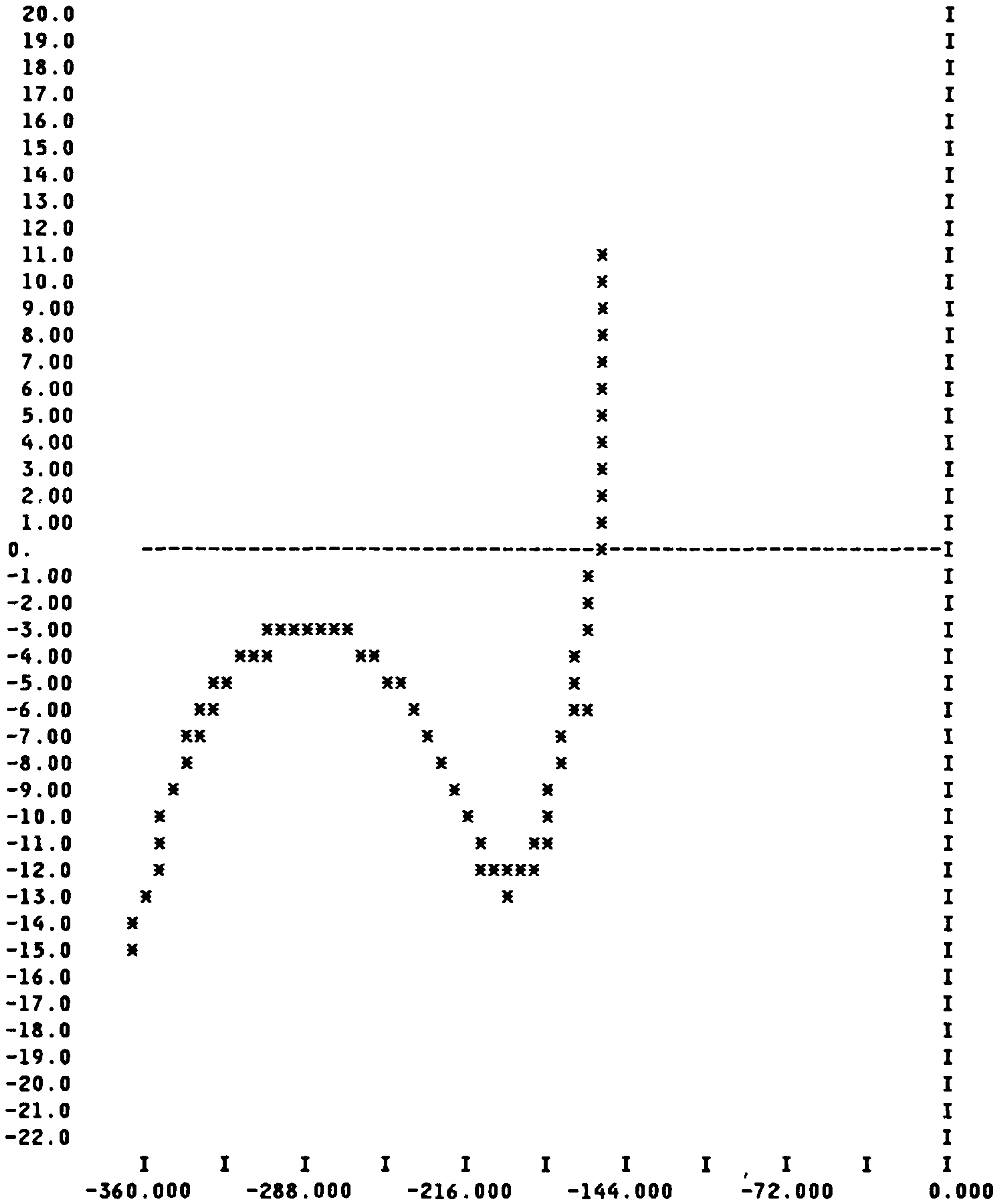
AUTOMATIC FREQUENCY MODE IF FAUTO.NE.0, FAUTO = 1.000

NOMEG = 2.000   OMEGA = 1.000   , 13.00   ,

OMEGA RAD/SEC	ZREAL	ZIMAG	REAL	IMAGINARY	DB	PHASE	PHASE MARGIN
1.000	.971	.238	-.331E+01	-.143E+01	11.125	-156.66	23.34
1.200	.959	.284	-.247E+01	-.113E+01	8.664	-155.48	24.52
1.400	.944	.330	-.195E+01	-.906E+00	6.648	-155.08	24.92
1.700	.918	.397	-.147E+01	-.670E+00	4.171	-155.51	24.49
2.000	.887	.462	-.117E+01	-.504E+00	2.125	-156.75	23.25
2.369	.843	.538	-.933E+00	-.360E+00	-.000	-158.91	21.09
2.869	.772	.635	-.722E+00	-.230E+00	-2.411	-162.36	17.64
3.469	.673	.740	-.560E+00	-.132E+00	-4.805	-166.75	13.25
4.269	.519	.854	-.423E+00	-.556E-01	-7.377	-172.53	7.47
5.069	.347	.938	-.339E+00	-.112E-01	-9.387	-178.11	1.89
5.346	.284	.959	-.317E+00	.458E-07	-9.966	-180.00	-.00
6.946	-.096	.995	-.241E+00	.467E-01	-12.206	-190.98	-10.98
8.546	-.462	.887	-.225E+00	.989E-01	-12.202	-203.76	-23.76
9.546	-.660	.752	-.250E+00	.181E+00	-10.213	-215.89	-35.39
10.05	-.745	.667	-.270E+00	.286E+00	-8.096	-226.62	-46.62
10.35	-.791	.612	-.262E+00	.409E+00	-6.287	-237.50	-57.50
10.55	-.819	.573	-.208E+00	.530E+00	-4.893	-248.60	-68.60
10.70	-.839	.544	-.110E+00	.630E+00	-3.887	-260.08	-80.08
10.82	-.855	.518	.250E-01	.688E+00	-3.247	-272.08	-92.08
10.92	-.868	.497	.160E+00	.690E+00	-2.994	-283.03	-103.03
11.02	-.879	.476	.292E+00	.640E+00	-3.054	-294.53	-114.53
11.12	-.890	.455	.393E+00	.546E+00	-3.442	-305.70	-125.70
11.22	-.901	.434	.447E+00	.435E+00	-4.097	-315.81	-135.81
11.37	-.916	.401	.458E+00	.283E+00	-5.376	-328.22	-148.22
11.57	-.934	.357	.409E+00	.150E+00	-7.222	-339.91	-159.91
11.82	-.954	.300	.335E+00	.654E-01	-9.326	-348.97	-168.97
12.12	-.973	.231	.270E+00	.236E-01	-11.343	-355.00	-175.00
12.62	-.994	.112	.212E+00	.339E-02	-13.460	-359.09	-179.09
13.00	-1.000	.022	.198E+00	.333E-03	-14.077	-359.90	-179.90

NICHOLS PLOT (MAGN. VS PHASE) 10/31/83

EXAMPLE 10 RATIONAL REPRESENTATION OF FREQUENCY DECOMPOSITION METHOD



FILM PLOT COMPLETED

Note that the frequency response is identical to the response computed in Example 9. In the tabulation of the response, the z-plane frequencies are at the slower sampling rate.

EXAMPLE 11 TRANSFER FUNCTION EVALUATION BY CRAMER'S METHOD

Problem: Given

$$\underline{M}(s) = \begin{vmatrix} s + 2 & -(s + 3) \\ 0 & .01 s^2 + .15 s + 1 \end{vmatrix}$$

$$\underline{B}(s) = \begin{vmatrix} 0 \\ 1 \end{vmatrix} \quad \text{and } \underline{X}(s) = \begin{vmatrix} x_1(s) \\ x_2(s) \end{vmatrix}$$

find the transfer function $x_1(s)/u(s)$ by application of Cramer's method.

$$H(s) = \frac{x_1(s)}{u(s)} = \frac{\det \underline{M}_1(s)}{\det \underline{M}(s)}$$

where $\underline{M}_1(s)$ is equal to $\underline{M}(s)$ with column 1 replaced by $\underline{B}(s)$.

Three operations are required for this example. The first two compute the determinants and store the results in polynomials and the last one copies the polynomials into an s-plane transfer function.

The FORTRAN code for this example is:

```
C     EXAMPLE 11
C     MUST INITIALIZE MATRIX PARAMETERS FOR FIRST TIME USE
C
      CALL MINITO
C     NOTE - DO NOT CALL MINITO AGAIN
C     .  .  .  .  .  .  .
C     ENTER PARAMETERS FOR DIMENSION AND DEGREE OF MATRIX
C
      MXM=2                 "dimension of matrices"
      MDEG=2                "highest degree of polynomial element"
C     ENTER MATRIX DATA FOR COEFFICIENTS OF S**0
      MO(1,1)=2
```

```
      M0(1,2)=-3
      M0(2,2)=1
C     ENTER MATRIX DATA FOR COEFFICIENTS OF S**1
      M1(1,1)=1
      M1(1,2)=-1
      M1(2,2)=.15
C     ENTER MATRIX DATA FOR COEFFICIENTS OF S**2
      M2(2,2)=.01
C     ENTER B VECTOR
      B0(2)=1.
C     COMPUTE DETERMINANT OF THE DENOMINATOR OF H(S)
      CALL REMARK5(
     +50HDETERMINANT OF DENOMINATOR WILL BE IN POLY2 AND    )
      CALL REMARK1(10HROOT2      )
      CALL DTERM(2,0)        "compute determinant of M(s)"
C                            "if 2nd argument is zero, no
C                             column substitution is made)
C     . . . . . . . .
C
C     COMPUTE DETERMINANT OF THE NUMERATOR OF H(S)
      CALL REMARK5(
     +50HDETRMINANT OF NUMERATOR WILL BE IN POLY3 AND      )
      CALL REMARK1(10HROOT3      )
      CALL DTERM(3,1)        "compute determinant of M (s)"
                                                      1
C                            "2nd argument is column number
C                             where B(s) is substituted"
C     . . . . . . . .
C
C     COPY POLYNOMIALS INTO S PLANE TRANSFER FUNCTION
C
      CALL CPYPS(13,3,2)
C     . . . . . . . .
```

The printer output for this example is:

```
DETERMINANT OF DENOMINATOR WILL BE IN POLY2 AND

ROOT2

MATRIX M(S) is

                    0              1              2              3
ROW  COL           S              S              S              S

 1    1      .20000000E+01   .10000000E+01
 1    2     -.30000000E+01  -.10000000E+01
 2    2      .10000000E+01   .15000000E+00   .10000000E-01
```

B VECTOR IS

ROW	S^0	S^1	S^2	S^3
2	.10000000E+01			

```
*************************************************************
*     DETRM - FIND DETERMINANT OF MATRIX                    *
*             WITH NO COLUMN REPLACED BY B VECTOR           *
*************************************************************

*************************************************************
*     ROOT2  = ROOTS OF DETERMINANT                         *
*************************************************************
```

THE ROOTS OF ROOT2 ARE

NO.	REAL	IMAG.	OMEGA	ZETA
1	-7.5000000	6.61437828	10.0000000	.750000000
2	-7.5000000	-6.61437828	10.0000000	750000000
3	-2.0000000	0.		

LOW ORDER NON ZERO COEFFICIENT =   2.0000000

```
*************************************************************
*     POLY2  = COEFFICIENTS OF DETERMINANT POLYNOMIAL       *
*************************************************************
```

DEGREE OF POLY2 IS 3   (COEFFICIENTS IN ASCENDING ORDER)
2. 1.3 .17 .01

DETERMINANT OF NUMERATOR WILL BE IN POLY3 AND

ROOT3

MATRIX M(S) is

ROW	COL	S^0	S^1	S^2	S^3
1	1	.20000000E+01	.10000000E+01		
1	2	-.30000000E+01	-.10000000E+01		
2	2	.10000000E+01	.15000000E+00	.10000000E-01	

```
B VECTOR IS

                  0                   1                   2                   3
ROW              S                   S                   S                   S

 2        .10000000E+01


************************************************************
*     DETRM - FIND DETERMINANT OF MATRIX                   *
*             WITH COLUMN  1 REPLACED BY B VECTOR          *
************************************************************

************************************************************
*       ROOT3  = ROOTS OF DETERMINANT                      *
************************************************************

                           THE ROOTS OF ROOT3 ARE

NO.         REAL             IMAG.             OMEGA              ZETA

 1     -3.0000000       0.

                    LOW ORDER NON ZERO COEFFICIENT =     [illegible]000000

************************************************************
*     POLY3  = COEFFICIENTS OF DETERMINANT OF POLYNOMIAL   *
************************************************************

DEGREE OF POLY3  IS  1          (COEFFICIENTS IN ASCENDING ORDER)
3. 1.


                           THE ROOTS OF ROOT3 ARE

NO.         REAL             IMAG.             OMEGA              ZETA

 1     -3.0000000       0.

                    LOW ORDER NON ZERO COEFFICIENT =   3.0000000

DEGREE OF POLY3  IS  1          (COEFFICIENTS IN ASCENDING ORDER)
3. 1.
```

THE ROOTS OF ROOT2 ARE

NO.	REAL	IMAG.	OMEGA	ZETA
1	-7.5000000	6.61437828	10.0000000	.750000000
2	-7.5000000	-6.61437828	10.0000000	750000000
3	-2.0000000	0.		

LOW ORDER NON ZERO COEFFICIENT = 2.0000000

DEGREE OF POLY2 IS 3 (COEFFICIENTS IN ASCENDING ORDER)
2. 1.3 .17 .01

```
*********************************************
*     SROOT13 =     ROOT3     /     ROOT2     *
*********************************************

*********************************************
*     SPTF13 =      POLY3     /     POLY2     *
*********************************************
```

THE NUMERATOR ROOTS OF SROOT13 ARE

NO.	REAL	IMAG.	OMEGA	ZETA
1	-3.0000000	0.		

LOW ORDER NON ZERO COEFFICIENT = 3.0000000

THE DENOMINATOR ROOTS OF SROOT13 ARE

NO.	REAL	IMAG.	OMEGA	ZETA
1	-7.5000000	6.61437828	10.0000000	.750000000
2	-7.5000000	-6.61437828	10.0000000	750000000
3	-2.0000000	0.		

LOW ORDER NON ZERO COEFFICIENT = 2.0000000

DEGREE OF NUMERATOR OF SPTF13 IS 1 (COEFFICIENTS IN ASCENDING ORDER)
3. 1.

DEGREE OF DENOMINATOR OF SPTF13 is 3 (COEFFICIENTS IN ASCENDING ORDER)
2. 1.3 .17 .01

BODE GAIN = 1.5000000

CP= 6.12

EXAMPLE 12 STORE DATA FROM BATCH JOB FOR LATER USE

Problem: Store all polynomial, transfer function, and matrix data so that it can be used in a subsequent batch or interactive LCAP2 job. Assume that Examples 1 through 11 were all run on the same batch job.

The LCAP2 STORE operator will store all pertinent data from a current job so that on subsequent batch or interactive jobs this data can be restored with the RESTORE operator. This data is written on TAPE31. At the completion of the batch job the user must catalog this file.

To identify the data on TAPE31, the first record will contain 70 characters of alphanumeric data copied from words HEAD(64) through HEAD(70). The user must enter this data before calling STORE.

The FORTRAN code for this example is:

```
C     EXAMPLE 12
C     ENTER INFORMATION TO IDENTIFY THE TAPE BEING WRITTEN ON

      CALL HEADIN5(64,
     +50HEXAMPLES FOR BATCH LCAP2 USERS MANUAL              )
      CALL HEADIN2(69,20HOCTOBER 31,1983    )
C
C     THE ABOVE CODE IS THE SAME AS
C     HEAD(64)=10HEXAMPLES F
C     HEAD(65)=10HOR BATCH L
C        .          .
C        .          .
C     HEAD(69)=10HOCTOBER 31
C     HEAD(70)=10H,1983
C
      CALL STORE(1)         "if argument is 0 printout will be suppressed"
      CALL LEXIT
      END
```

The printer output for this example is[1] :

---

[1] Since the batch job represented by Examples 1 through 11 generated nineteen different data sets, for brevity only a representative sample of the output for this example is given here.

```
***************************************************
*      STORE - STORE POLYNOMIALS, TRANSFER FUNCTIONS,     *
*             AND MATRIX DATA FOR A FUTURE JOB            *
***************************************************

TAPE31 IDENTIFIER IS

EXAMPLES FOR BATCH LCAP2 USERS MANUAL          OCTOBER 31,1983

                         THE ROOTS OF ROOT1  ARE

NO.        REAL            IMAG.           OMEGA           ZETA
 1      -1.8445105      -.49938380       1.9109168       .96524896
 2      -1.8445105       .49938380       1.9109168       .96524896
 3      -9.3109790      0.

DEGREE OF POLY1  IS  3          (COEFFICIENTS IN ASCENDING ORDER)
34. 38. 13. 1.
                         THE ROOTS OF ROOT2  ARE

NO.        REAL            IMAG.           OMEGA           ZETA

 1      -7.5000000       6.6143783       10.000000       .75000000
 2      -7.5000000      -6.6143783       10.000000       .75000000
 3      -2.0000000      0.

DEGREE OF POLY2  IS  3          (COEFFICIENTS IN ASCENDING ORDER)
2. 1.3 .17 .01
                                .
                      (data for polynomial 3)
                                .
                                .
              (data for s-plane transfer functions 1,...,5)
                                .
                                .
               (data for z-plane transfer functions 1,2,3)
                                .
                                .
              (data for s-plane transfer functions 6,...,13)
                                .
                                .
MXM = 2
MDEG = 2
```

MATRIX M(S) is

ROW	COL	S^0	S^1	S^2	S^3
1	1	.20000000E+01	.10000000E+01		
1	2	-.30000000E+01	-.10000000E+01		
2	2	.10000000E+01	.15000000E+00	.10000000E-01	

B VECTOR IS

ROW	S^0	S^1	S^2	S^3
2	.10000000E+01			

Note that the s-plane transfer functions are not listed continuously from 1 through 13. This is due to the method in which polynomials and transfer functions are stored in the program. All polynomials and transfer functions with indices 1 through 5 are saved in SCM (small core memory). All polynomials and transfer functions with indices greater than 5 are saved on disk storage. In printing out the data stored by the STORE operator, the data which is saved in SCM is printed out before the data which is saved on disk storage.

After execution of Batch LCAP2 has been completed, the job will proceed according to the operations specified in the control card input file in Section 5.0. First, file TAPE31 will be cataloged. For this example the file name used is 8LCAP2BATCHDATA with ID=9487. This file will be used by Example 13. Then, the hardcopy plots will be processed by the program HARDCPY.

EXAMPLE 13 RESTORE DATA FROM A PREVIOUS BATCH LCAP2 SESSION

Problem: Restore data stored in Example 12

In Example 12, the STORE operator was used to save data on file TAPE31. Upon completion of the batch job this file was stored as file 8LCAP2BATCHDATA, ID=9487. To restore this data for a new batch job this file must first be attached as file TAPE30 before LCAP2 is executed.

The job structure for this example is:

```
      .
      .            (control cards for accounting)
      .
FILE,TAPE30,BT=I.
ATTACH(TAPE30,8LCAP2BATCHDATA,ID=9487,ST=PF6)
ATTACH(X,8LCAP2CC,ID=9487)
BEGIN,LCAP2CC,X.
*EOR
*IDENT idname
*DECK MAIN
*CALL LCAP2
C
      CALL INITO
      CALL MINITO
           .
           .
           .
      (user's FORTRAN code)
           .
      CALL LEXIT
      END
```

The user's FORTRAN code for this example is:

```
C      RESTORE DATA STORED IN EXAMPLE 12
C
C      CALL RESTORE(1)         "if argument is 0 printout will be suppressed"
```

The printer output for this example is[1] :

```
*************************************************************
*    RESTORE - RESTORE POLYNOMIALS, TRANSFER FUNCTIONS,     *
*              AND MATRIX DATA FROM A PREVIOUS JOB          *
*************************************************************

TAPE30 IDENTIFIER IS

EXAMPLES FOR BATCH LCAP2 USERS MANUAL          OCTOBER 31,1983

                         THE ROOTS OF ROOT1  ARE

NO.        REAL           IMAG.            OMEGA          ZETA
 1     -1.8445105     -.49938380       1.9109168      .96524896
 2     -1.8445105      .49938380       1.9109168      .96524896
 3     -9.3109790     0.

DEGREE OF POLY1  IS  3          (COEFFICIENTS IN ASCENDING ORDER)
34. 38. 13. 1.
                         THE ROOTS OF ROOT2  ARE

NO.        REAL           IMAG.            OMEGA          ZETA

 1     -7.5000000      6.6143783       10.000000      .75000000
 2     -7.5000000     -6.6143783       10.000000      .75000000
 3     -2.0000000     0.

DEGREE OF POLY2  IS  3          (COEFFICIENTS IN ASCENDING ORDER)
2. 1.3 .17 .01
                                 .
                      (data for polynomial 3)
                                 .
                                 .
             (data for s-plane transfer functions 1,...,5)
                                 .
                                 .
              (data for z-plane transfer functions 1,2,3)
                                 .
                                 .
             (data for s-plane transfer functions 6,...,13)
                                 .
                                 .
```

[1] Since the batch job represented by Examples 1 through 11 generated nineteen different data sets, for brevity only a representative sample of the output for this example is given here.

MXM = 2
MDEG = 2

MATRIX M(S) is

ROW	COL	S^0	S^1	S^2	S^3
1	1	.20000000E+01	.10000000E+01		
1	2	-.30000000E+01	-.10000000E+01		
2	2	.10000000E+01	.15000000E+00	.10000000E-01	

B VECTOR IS

ROW	S^0	S^1	S^2	S^3
2	.10000000E+01			

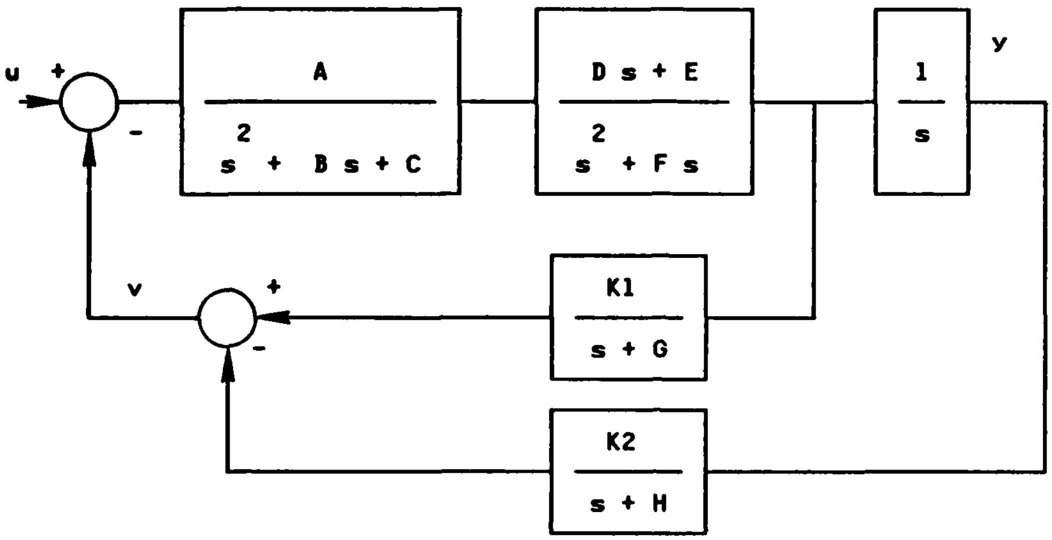
9.0 MORE EXAMPLES

In Section 8 examples were presented for some commonly used LCAP2 operators. The FORTRAN code and the printer output for each example were presented. In this section more examples covering advanced operations will be presented. However, no printer output will be included.

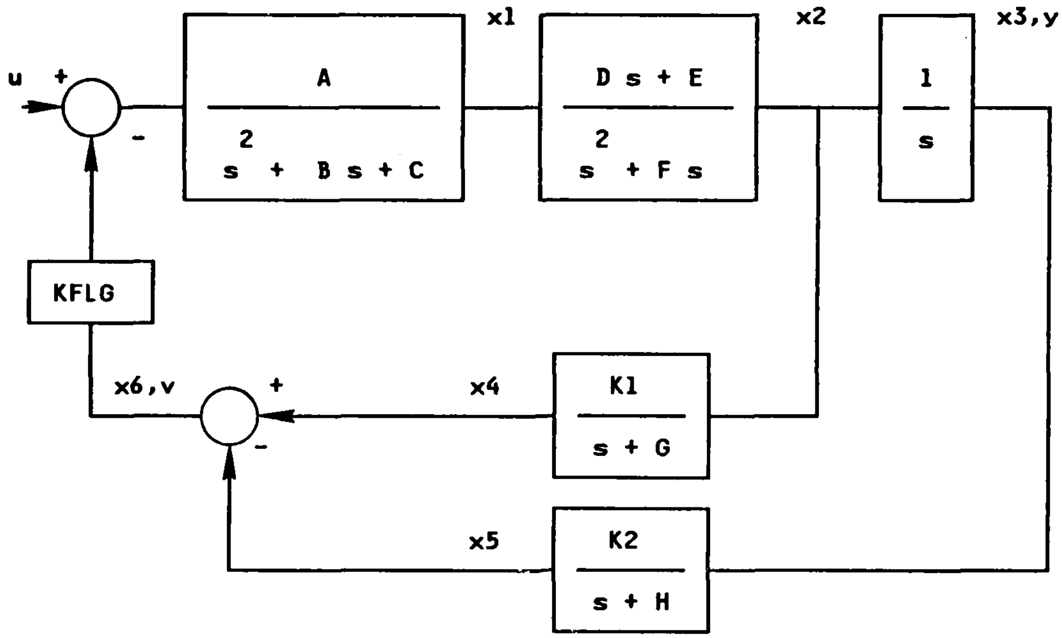
Two examples are included. The first one describe the use of Cramer's method for transfer function evaluation. The second one describes the use of a user-supplied function to compute the s-plane frequency response of a non-rational function. This technique can also be utilized for rational functions which exceed LCAP2's restriction that transfer functions be less than degree 50. The frequency response of two transfer functions, each whose degree is less that 50, but whose sum is greater than 50, can be computed in a similar manner as that described in Example 15.

EXAMPLE 14 MATRIX REPRESENTATION OF AN S-PLANE SYSTEM

Problem: Compute the closed loop transfer function y/u and the open loop transfer function v/u of the following system using Cramer's method.



For stability analysis the loop is to be broken after the summation of the two feedback loops. Defind a flag KFLG, whose value is to be 0 or 1, which will be used to open or close this feedback loop. Redraw the above figure with his flag and with state variables x1, through x6 to obtain



The state equations are,

$$(s^2 + B s + C) (x1) = (A) (u - (KFLG) (x6))$$

$$(s^2 + F s) (x2) = (D s + E) (x1)$$

$$(s) (x3) = x2$$

$$(s + G) (x4) = (K1) (x2)$$

$$(s + H) (x5) = (K2) (x3)$$

$$(x6) = x4 + x5$$

The matrix form of these equations are

$$\underline{M}(s) * \underline{X}(s) = \underline{B}(s) * u$$

where $\underline{M}(s)$ is given as,

$$\underline{M}(s) = \begin{vmatrix} s^2 + Bs + C & 0 & 0 & 0 & 0 & KFLG \\ -Ds - E & s^2 + Fs & 0 & 0 & 0 & 0 \\ 0 & -1 & s & 0 & 0 & 0 \\ 0 & -K1 & 0 & s+G & 0 & 0 \\ 0 & 0 & -K2 & 0 & s+H & 0 \\ 0 & 0 & 0 & -1 & -1 & 1 \end{vmatrix}$$

and $\underline{B}(s)$ is given by

$$\underline{B}(s) = (A,0,0,0,0,0)^T$$

The closed loop transfer function y/u is given by

$$\frac{y(s)}{u(s)} = \frac{x_3(s)}{u(s)} = \frac{\det \underline{M}_3(s)}{\det \underline{M}(s)}$$

where $\underline{M}_3(s)$ is equal to $\underline{M}(s)$ with column 1 replaced by $\underline{B}(s)$ and the value of KFLG = 1.

The open loop transfer function v/u is given by

$$\frac{v(s)}{u(s)} = \frac{x_6(s)}{u(s)} = \frac{\det \underline{M}_6(s)}{\det \underline{M}(s)}$$

where $\underline{M}_6(s)$ is equal to $\underline{M}(s)$ with column 6 replaced by $\underline{B}(s)$ and the value of KFLG = 0.

To use LCAP2 to evaluate the above transfer functions, the polynomial matrix $\underline{M}(s)$ is represented by the input matrices $\underline{M0}$, $\underline{M1}$, $\underline{M2}$, $\underline{M3}$, and $\underline{M4}$ as

$$\underline{M}(s) = \underline{M4}\, s^4 + \underline{M3}\, s^3 + \underline{M2}\, s^2 + \underline{M1}\, s^1 + \underline{M0}$$

The $\underline{Mi}$, i=1,...,4 input matrices should not be confused with the $\underline{M}_i(s)$ matrix used in computing the numerator of a transfer function.

The FORTRAN code for this example is:

```
      REAL K1,K2,KFLG
C
C     MUST INITIALIZE MATRIX PARAMETERS FOR FIRST TIME USE
C
      CALL INITO
      CALL MINITO
C     NOTE - DO NOT CALL MINITO AGAIN
C
C     ENTER PARAMETERS FOR ORDER AND DEGREE OF MATRIX
C
      MXM=6                      "degree of matrix"
      MDEG=2                     "highest degree of poly. element"
C
C     ENTER DATA
      A=.....
      B=.....
      C=.....
      D=.....
      E=.....
      F=.....
      G=.....
      H=.....
      K1=.....
      K2=.....
C     ENTER DATA FOR CLOSED LOOP CASE FIRST
      M0(1,1)=C
      M0(1,6)=KFLG=1.            "set KFLG for closed loop"
      M0(2,1)=-E
      M0(3,2)=-1
      M0(4,2)=-K1
      M0(4,4)=G
      M0(5,3)=-K2
      M0(5,5)=H
      M0(6,4)=-1
      M0(6,5)=-1
      M0(6,6)=1
      M1(1,1)=B
      M1(2,1)=-D
      M1(2,2)=F
      M1(3,3)=1
      M1(4,4)=1
      M1(5,5)=1
      M2(1,1)=1
      M2(2,2)=1
      B0(1)=A
C     COMPUTE DENOMINATOR OF CLOSED LOOP
C
      CALL DTERM(1,0)            "denom. det. of closed loop"
```

```
C     COMPUTE NUMERATOR OF CLOSED LOOP
C
      CALL DTERM(2,3)           "num. det. of closed loop"
C
C     COPY NUM. AND DENOM. INTO CLOSED LOOP TRANSFER FUNCTION
C
      CALL CPYPS(1,2,1)         "SPTF1 is the closed loop t.f."
C
C     CHANGE ELEMENTS FOR OPEN LOOP CONFIGURATION
C
      M0(1,6)=KFLG=0            "set KFLG for open loop"
C
C     COMPUTE DENOMINATOR OF OPEN LOOP
C
      CALL DETRM(3,0)           "denom. det. for open loop"
C
C     COMPUTE NUMERATOR OF OPEN LOOP
C
      CALL DETRM(4,6)           "num. det. for open loop"
C
C     COPY NUM. AND DENOM. INTO OPEN LOOP TRANSFER FUNCITON
C
      CALL CPYPS(2,4,3)         "SPTF2 is the open loop t.f."
C
C     DESIRED TRANSFER FUNCTIONS HAVE BEEN COMPUTED
C
-----------------------------------------------------------------------------
```

EXAMPLE 15 S-PLANE FREQUENCY RESPONSE OF A NON RATIONAL FUNCTION

Problem: Find the s-plane transfer function of the non rational[1] function

$A(s) + B(s)\,e^{-ds}$ where A(s) and B(s) are rational transfer functions stored in LCAP2 as SPTF1 and SPTF2, respectively.

The s-plane operator SFREQ(i) has a provision to include time delay by cascading the response of SPTFi with time delay specified by FDLAY. It is not capable though, of handling a more general non rational function such as the one for this example.

The operator FREQS(SFAUX1), which uses a user supplied COMPLEX FUNCTION SFAUX1, can be used to obtain the desired frequency response. This operator and function pair has been designed so that the code to be written by the user can be simple. The user-supplied function will enable the user to (1) compute the response of A(s) and B(s) individually, (2) compute the response of e^{-ds} and (3) combine the results so that $A(s) + B(s)\,e^{-ds}$ is the function to be evaluated.

The user UPDATE code for this example is:

```
*IDENT idname
*INSERT START.1
*DECK MAIN
*CALL LCAP2
      EXTERNAL SFAUX1         "must be before first executable
                               statement"
      CALL INITO              "initialization of LCAP2 param."
            .
            .
      code defining SPTF1
      and SPTF2
            .
            .
      frequency response parameters,
      i.e., FAUTO, NOMEG, OMEGA(1),
      FBODE, etc.
            .
            .
      CALL FREQS(SFAUX1)      "SFAUX1 is the user-supplied
                               COMPLEX FUNCTION"
```

---

[1] This non rational function could represent a distributed parameter system.

```
      CALL LEXIT
      END
C
C     THE FOLLOWING ARE UPDATE CHANGES TO COMPLEX FUNCTION SFAUX1
C
C     COMPLEX FUNCTION SFAUX1 IS IN THE LCAP2 SUBROUTINE LIBRARY.
C     USER NEEDS ONLY TO CHANGE STATEMENT 100 TO ADD CODE TO
C     DEFINE HIS/HER PARTICULAR PROBLEM.
C
C     USER NEED NOT BE CONCERNED WITH PASSING THE ARGUMENTS FOR
C     COMPUTING THE FREQUENCY RESPONSE. IT WILL BE DONE AUTO-
C     MATICALLY.
*DELETE SFAUX.37             "SFAUX.37 is the card ident for
                              statement 100"
      D=.....                "value of delay d"
      DS=-D*U                "value of ds"
  100 SFAUX1=SFAUX(SPTF1)+SFAUX(SPTF2)*CEXP(CMPLX(0.,DS))
C
C     EXPLANATION OF STATEMENT 100 -
C     SFAUX(SPTF1) IS A COMPLEX VALUE EQUAL TO THE RESPONSE AT
C     COMPLEX FREQUENCY U, WHERE U OF COMMON/FRQBLK/ IS THE S
C     PLANE OMEGA FREQUENCY BEING VARIED AUTOMATICALLY BY THE
C     PROGRAM. CEXP(CMPLX(0.,DS)) IS THE COMPLEX VALUE OF THE
C     DELAY. SFAUX1 IS THE RETURNED VALUE OF THE RESPONSE OF
C     THE SPECIFIED FUNCTION.
C
```

So that changes to the user-supplied COMPLEX FUNCTION SFAUX1 can be understood, the code for this routine has been included in the description of SFAUX1 in Appendix C.

The code for COMPLEX FUNCTION SFAUX1 has been set up so that the user can access the first five s, z and w plane transfer functions without additional changes to the code. If other transfer functions are to be used, see description of this routine in Ref. 2.

REFERENCES

1. E. A. Lee, "Linear Controls Analysis Program (LCAP) Users Guide," Aerospace Corporation, TOR - 0077(2442-23)-1, 5 October 1976.

2. E. A. LEE, "LCAP2 - Linear Controls Analysis Program, Vol III: Source Code Description," Aerospace Corporation, TR - 0084(9975)-1 Vol III, 15 November 1983.

3. E. A. LEE, "LCAP2 - Linear Controls Analysis Program, Vol II: Interactive LCAP2 User's Guide," Aerospace Corporation, TR - 0084(9975)-1 Vol II, 15 November 1983.

4. E. A. Lee, "LCAP2-Linear Controls Analysis Program," IEEE Control Systems Magazine, December 1982, pp. 15-18.

APPENDIX A - DESCRIPTION OF LCAP2 OPERATORS

LCAP2 operators are FORTRAN subroutines which are written to allow the user to easily perform control system analysis operations using only a few arguments. In most cases the arguments will be indices defining polynomials and/or transfer functions to be operated upon. Descriptions of these operators are presented in alphabetical order.

The degree of the polynomials and transfer functions referenced and generated by these LCAP2 operators must be less than 50.

In addition to these operators, the user can utilize other LCAP2 subroutines and functions which are described in Appendix C.

CPYPS

Identification

SUBROUTINE CPYPS - LCAP2 Operator, Copy Polynomials Into S Plane Transfer Function

Purpose

Copy polynomials into an s plane transfer functions using LCAP2 indices. For transfer function evaluation by Cramer's method, this operator is used to define a transfer function after two polynomial determinants have been computed with the use of the DTERM or DETRM operators.

Usage

```
CALL CPYPS(I,J,K)
```

I input - Index of s plane transfer function where results are to be stored
J input - Index of poly. to be used to define numerator of SPTFi, i=I
K input - Index of poly. to be used to define denominator of SPTFi, i=I

Method

Copies coefficients of polynomials into a transfer function. If the roots of the polynomials are also defined, these roots are also copied into the transfer function.

PREVIOUS PAGE IS BLANK

CPYPW

Identification

SUBROUTINE CPYPW - LCAP2 Operator, Copy Polynomials Into W Plane Transfer Function

This operator is similar to CPYPS except that it is for use with a w plane transfer function instead of an s plane transfer function.

CPYPZ

Identification

SUBROUTINE CPYPZ - LCAP2 Operator, Copy Polynomials Into Z Plane Transfer Function

This operator is similar to CPYPS except that it is for use with a z plane transfer function instead of an s plane transfer function.

CPYSP

Identification

SUBROUTINE CPYSP - LCAP2 Operator, Copy S Plane Transfer Function Into Polynomials

Purpose

Copy s plane transfer function into polynomials using LCAP2 indices.

Usage

CALL CPYSP(I,J,K)

I input - Index of s plane transfer function to be used in copying
J input - Index of polynomial equated with the numerator of SPTFi, i=I
K input - Index of polynomial equated with the denominator of SPTFi, i=I

Method

Copies coefficients of transfer function into polynomials. If the roots of the transfer function are available, the roots are also stored in the polynomials.

CPYWP

Identification

SUBROUTINE CPYWP - LCAP2 Operator, Copy W Plane Transfer Function Into Polynomials

This operator is similar to CPYSP except that it is for use with a w plane transfer function instead of an s plane transfer function.

CPYZP

Identification

SUBROUTINE CPYZP - LCAP2 Operator, Copy Z Plane Transfer Function Into Polynomials

This operator is similar to CPYSP except that it is for use with a z plane transfer function instead of an s plane transfer function.

DTERM

Identification

SUBROUTINE DTERM - LCAP2 Operator, Determinant Of Matrix With Polynomial Elements (New Version)

Purpose

Transfer function evaluation by Cramer's method for the system described by

$$\underline{M}(s)\ \underline{X}(s) = \underline{B}(s)u$$

where $\underline{M}(s) = \underline{M4}\, s^4 + \underline{M3}\, s^3 + \underline{M2}\, s^2 + \underline{M1}\, s^1 + \underline{M0}$

$$\underline{B}(s) = \underline{B4}\, s^4 + \underline{B3}\, s^3 + \underline{B2}\, s^2 + \underline{B1}\, s^1 + \underline{B0}$$

$\underline{M0}$, $\underline{M1}$, $\underline{M2}$, $\underline{M3}$, $\underline{M4}$ are square matrices of dimension MXM
$\underline{B0}$, $\underline{B1}$, $\underline{B2}$, $\underline{B3}$, $\underline{B4}$ are vectors of dimension MXM

$\underline{X}(s)$ = State vector of dimension MXM
u = Scalar input

is given by

$$\frac{x_j(s)}{u} = \frac{\det \underline{M}_j(s)}{\det \underline{M}(s)}$$

where $\underline{M}_j(s)$ is equal to $\underline{M}(s)$ with column j replaced by $\underline{B}(s)$.

The operator DTERM will compute the determinant of $\underline{M}_j(s)$. Substitution of $\underline{B}(s)$ into column j will be done automatically by the program.

The determinant is found by solving for its roots directly and then computing its coefficients.

Usage

CALL DTERM(I,J)

I input - Index where polynomial determinant is to be stored
J input - Column where $\underline{B}(s)$ is to be substituted into
(J=0 interpreted to mean no column substitution)

1. Before DTERM is used, the matrix parameters must first be initialized by calling MINIT0 (only once).

2. Matrix parameters are in COMMON/MATRIX1/. They are to be set before MROOT1 is called. These parameters are defined below:

Parameters	Preset	Description
MXM	1	Dimension of matrices and vectors (1-30)
MDEG	0	Highest degree of polynomial element (0-4)
M0	0	Matrix for coefficients of s**0
M1	0	Matrix for coefficients of s**1
M2	0	Matrix for coefficients of s**2
M3	0	Matrix for coefficients of s**3
M4	0	Matrix for coefficients of s**4
B0	0	Vector for coefficients of s**0
B1	0	Vector for coefficients of s**1
B2	0	Vector for coefficients of s**2
B3	0	Vector for coefficients of s**3
B4	0	Vector for coefficients of s**4

Method

If j is not zero, $\underline{B}(s)$ is substituted into column j of $\underline{M}(s)$. Subroutine MROOT1 is then called to compute the determinant. Column j of $\underline{M}(s)$ is then restored to its original value.

Restrictions

The dimension of the matrix must not be greater than 30 x 30. The polynomial elements of the matrix must be of degree 4 or less. The degree of the computed polynomial determinant must be less than 50.

FREQS

Identification

SUBROUTINE FREQS - LCAP2 Operator, Frequency Response Using A User-Supplied Function

Purpose

Compute frequency response of an arbitrary s plane transfer function.

Usage

```
CALL FREQS(FAUX1)
```

FAUX1 input - Name of user supplied subroutine. Must be declared with an EXTERNAL statement in the calling program.

1. Frequency response parameters are in COMMON/HEADDB/. See description of SFREQ. The user need not be concerned with passing the arguments for computing the frequency response. It is automatically done by LCAP2.

FREQW

Identification

SUBROUTINE FREQW - LCAP2 Operator, W Plane Frequency Response Using A User-Supplied Function

Purpose

Compute w plane frequency response of an arbitrary w plane transfer function.

Usage

```
CALL FREQW(FAUX1)
```

FAUX1 input - Name of user supplied subroutine. Must be declared with an EXTERNAL statement in the calling program.

1. Frequency response parameters are in COMMON/HEADDB/. See description of SFREQ. The user need not be concerned with passing the arguments for computing the frequency response. It is automatically done by LCAP2.

FREQZ

Identification

SUBROUTINE FREQZ - LCAP2 Operator, Z Plane Frequency Response Using A User-Supplied Function

Purpose

Compute frequency response of an arbitrary z plane transfer function.

Usage

CALL FREQZ(FAUX1)

FAUX1 input - Name of user supplied subroutine. Must be declared with an external statement in the calling program.

1. Frequency response parameters are in COMMON/HEADDB/. See description of SFREQ. The user need not be concerned with passing the arguments for computing the frequency response. It is automatically done by LCAP2.

PADD

Identification

SUBROUTINE PADD - LCAP2 Operator, Polynomial Add

Purpose

Add two polynomials using LCAP2 indices.

Usage

CALL PADD(I,J,K)

I input - Index of resultant polynomial sum
J input - Index of first polynomial to be added
K input - Index of second polynomial to be added

PEQU

Identification

SUBROUTINE PEQU - LCAP2 Operator, Polynomial Equal

Purpose

Equate polynomials using LCAP2 indices.

Usage

CALL PEQU(I,J)

I input - Index of resultant polynomial
J input - Index of polynomial to be equated with

PLDC

Identification

SUBROUTINE PLDC - LCAP2 Operator, Polynomial Load In Coefficient Form

Purpose

Load coefficients into polynomial coefficient array, POLYi.

Usage

CALL PLDC(I)

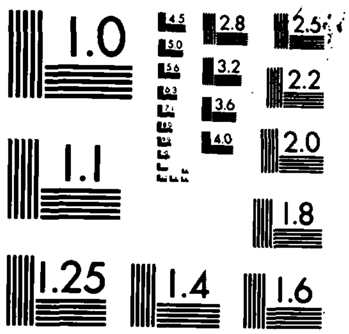
I input - Index where polynomial is to be stored

1. Polynomial coefficients are entered with polynomial array POLY (LCAP2 format) which are in COMMON/HEADDB/.

2. The calling program must include COMMON/HEADDB/ and the appropriate EQUIVALENCE and DIMENSION statements for POLY.

3. The roots of POLYi will not be automatically computed. If this is desired, follow this operation with the operator PRTS(I).

AD-A141 486 LCAP2 (LINEARCCONTROL ANALYSIS PROGRAM) VOLUME 1 BATCH 2/2
LCAP2 USER'S GUIDE. (U) AEROSPACE CORP EL SEGUNDO CA
GUIDANCE AND CONTROL DIV E A LEE 15 NOV 83
UNCLASSIFIED SD-TR-84-06-VOL-1 F04701-83-C-0084 F/G 9/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

PLDR

Identification

SUBROUTINE PLDR - LCAP2 Operator, Polynomial Load In Root Form

Purpose

Load roots in polynomial root array, ROOT. After the roots have been loaded, the coefficients of the polynomial are computed and stored in the polynomial coefficient array POLYi.

Usage

```
CALL PLDR(I)
```

I input - Index where polynomial is to be stored

1. Polynomial roots are entered with polynomial root array ROOT (LCAP2 format) which is in COMMON/HEADDB/.

2. The calling program must include COMMON/HEADDB/ and the appropriate EQUIVALENCE and DIMENSION statements for ROOT.

PMPY

Identification

SUBROUTINE PMPY - LCAP2 Operator, Polynomial Multiply

Purpose

Multiply two polynomials using LCAP2 indices.

Usage

```
CALL PMPY(I,J,K)
```

I input - Index of resultant polynomial product
J input - Index of polynomial multiplicand
K input - Index of polynomial multiplier

Method

If only the coefficients of the j-th and k-th polynomials are available, the product is computed by multiplication of the coefficients. If the roots of the j-th and k-th polynomials are available, the product is computed by combining the roots. The coefficients of the product are then formed from these roots.

PPRN

Identification

SUBROUTINE PPRN - LCAP2 Operator, Print Out Polynomial

Purpose

Print out a polynomial using an LCAP2 index.

Usage

```
CALL PPRN(I)
```

I input - Index of polynomial to be printed out

Method

Roots of the polynomial are printed out only if they are defined (previously computed or loaded in). Coefficients of the polynomial are printed out in ascending order.

PRTS

Identification

SUBROUTINE PRTS - LCAP2 Operator, Find Roots Of A Polynomial

Purpose

Find roots of a polynomial using an LCAP2 index.

Usage

```
CALL PRTS(I)
```

I input - Index of polynomial

Restrictions

If the roots of POLYi were previously computed or loaded in, the program will not recompute the roots from the coefficients. A message to this effect will be printed.

PSUB

Identification

SUBROUTINE PSUB - LCAP2 Operator, Polynomial Subtract

Purpose

Subtract two polynomials using LCAP2 indices.

Usage

CALL PSUB(I,J,K)

I input - Index of resultant polynomial difference
J input - Index of first polynomial (minuend)
K input - Index of second polynomial (subtrahend)

RESTORE

Identification

SUBROUTINE RESTORE - Restore Polynomial, Transfer Function And Matrix Data

Purpose

Restore polynomial, transfer function and matrix data from a previous batch or interactive job for a restart capability in Batch LCAP2.

Usage

CALL RESTORE(IPRNFG)

IPRNFG input - =0 for no printout of restored data

Restrictions

File type for TAPE30 must be declared with 'FILE,TAPE30,BT=I.' File TAPE30 must be attached before executing LCAP2.

SELCR

Identification

SUBROUTINE SELCR - LCAP2 Operator, Eliminate Common Roots Of S Plane Transfer Function

Purpose

Eliminate common roots from an s plane transfer function using an LCAP2 index.

Usage

```
CALL SELCR(I)
```

I input - Index of s plane transfer function

1. Common root elimination parameters ECRE1 (preset=2.E-4) and ECRE2 (preset=1.E-8) are in COMMON/HEADDB/.

Method

If a numerator root nrt and a denominator root drt are found such that ABS(drt/nrt - (1.,0.)).LT.ECRE1 for nrt.NE.0 or ABS(drt).LT.ECRE2 for nrt.EQ.0, roots nrt and drt are considered to be common roots and will be eliminated from the transfer function.

SFREQ

Identification

SUBROUTINE SFREQ - LCAP2 Operator, S Plane Frequency Response

Purpose

Evaluate s plane frequency response using an LCAP2 index. Automatic frequency mode available to allow program to dynamically choose its own frequency points to yield a smooth plot of the response.

Usage

CALL SFREQ(I)

I input - Index of s plane transfer function

1. Frequency response parameters are in COMMON/HEADDB. They are to be set before SFREQ is called. These parameters are defined below:

parameter	preset	description
FAUTO	1	.NE.0 for automatic frequency mode. Uses NOMEG and OMEGA array. .EQ.0 for user supplied frequency points. Uses FREQ1, FREQ2, ..., FREQ5 arrays.
NOMEG	2	Number of frequency points entered in OMEGA array for use in auto. frequency mode (max=20)
OMEGA		Array of frequency points for auto. freq. mode (Units defined by RAD).
	1.	OMEGA(1)=first frequency point used in auto. mode
	10.	OMEGA(2)=second frequency point used in auto. mode
		.
		.
		OMEGA(NOMEG)=last freq. point used in auto. mode
RAD	1	.NE.0 for rad/sec, .EQ.0 for HZ
FBODE	1	.NE.0 for Bode plot
FNICO	0	.NE.0 for Nichols plot
PMARG	0	.NE.0 for plotting phase margin instead of phase for the Nichols plot
FNYQS	0	.NE.0 for Nyquist plot
NQDB	0	.NE.0 for hardcopy Nyquist plot in db
GRAFP	1	.NE.0 for printer (lower resolution) plots
FILM	0	.NE.0 for hardcopy (higher resolution) plots
FDLAY	0	Time delay (dead time) (s plane only)
DEGMN	-360.	Minimum defined phase in frequency response (Phase defined from DEGMN to DEGMN+360.)
CYCLE	0	.EQ.0 for automatic selection of 2 or 3 cycle scale for Bode plots. (1 cycle not available)

FREQ1(1)	1	Starting freq. point for first segment of user specified values
FREQ1(2)	10	End freq. point for second segment of user specified values
FREQ1(3)	1	Delta frequency for third segment of user specified values
FREQk(1)	0	Starting freq. point for k-th segment *
FREQk(2)	0	End freq. point for k-th segment *
FREQk(3)	0	Delta frequency for k-th segment *
(k=2,5)		(* - only if FAUTO.EQ.0)
DBMAX	0	Maximum db for plotting frequency response
DBMIN	0	Minimum db for plotting frequency response (Auto. scaling if DBMAX=DBMIN)
FXYDL	.5	Nyquist plot scale in units per inch (Auto. scaling if FXYDL=0)
FXYMN	-2.5	Nyquist plot parameter - minimum real and imaginary value plotted.
SAMPT	1.	Sampling period for use in w or z plane freq. response

Method

If the automatic frequency mode is selected (FAUTO=0), the program will choose frequency values for evaluating the transfer function such that successive delta db and delta phase values will be within specified limits to yield a smooth plot. The program evaluates the first point using f = OMEGA(1). Then choosing deltaf = OMEGA(1)/20 initially, the next frequency to be used is computed as f = f + deltaf. Evaluating the next point using this value of f, the delta db and delta phase is compared to the specified limits. If either is too large, deltaf is halved and the response is recomputed. If both are too small, deltaf is doubled and the response is recomputed. The limits for the delta db response is EDB1/2 and EDB1. The limits for the delta phase response is EDEG1/2 and EDEG1. Simultaneously with computing the next f to be used, a comparison is made with the next value of OMEGA(i). If f is larger than OMEGA(i), f will be replaced with the value of OMEGA(i). This will ensure that the user specified frequency values will be inserted into those computed by the program. This process will continue until the last value OMEGA(NOMEG) is used.

Since the plot points computed to generate a smooth plot will, in many cases, be very large, only a portion of the computed response will be printed out. The print out is controlled by the delta db and delta phase parameters, EDB2 and EDEG2, respectively. A print out is made only if either of these two limits are exceeded.

Also as part of the automatic frequency mode, a comparison is made on deltaf to keep (deltaf/f) within the limits of MNDW and MXDW. The lower limit MNDW is necessary to prevent an excessive number of plot points around frequencies with low damping coefficients. The upper limit MXDW will ensure enough points to yield a smooth Bode plot.

The above parameters used in the automatic frequency mode are in COMMON/HEADDB/. They can be changed by the user. These parameters are defind below:

parameter	preset	description
EDB1	1.	Min. delta db for plotting
EDB2	2.	Min. delta db for printout
EDEG1	4.	Max. delta phase for plotting
EDEG2	10.	Max. delta phase for printout
MNDW	.0005	Min relative frequency step size
MXDW	.2	Max. relative frequency step size
MXITF	3000	Max. no. of iterations of freq. response

With either the automatic or the non-automatic frequency mode, the program will automatically check for the gain and phase crossover. When found, interpolation is used to find the exact crossover frequency and the response computed at that frequency.

In the non-automatic frequency mode (FAUTO=0) the user can define up to five sets of frequencies to be used in computing the response. Each of these sets are specified by a three element array of the form FREQk(i), i=1,3 described above under Usage. If FREQk(1) = a FREQk(2) = b and FREQk(3) = c, the k-th set for frequencies specified is:

a, a+c, a+2c, ..., a+jc, b

where j is the largest integer such that (a+jc) is less than b. Each successive FREQk array must define an increasing set of frequencies such that the first value of the segment is always larger than the last value of the preceding segment. When FREQk(3) is not larger than FREQk(1), as in the case with the preset values for k = 2,5, those segments will not be used.

SLOCI

Identification

SUBROUTINE SLOCI - LCAP2 Operator, S Plane Root Locus

Purpose

Evaluate s plane root locus using an LCAP2 index. Automatic gain selection available to supplement user-selected gains.

Usage

CALL SLOCI(I)

I input - Index of s plane transfer function to be evaluated

1. Root locus parameters are in COMMON/HEADDB/. They are to be set before SLOCI is called. These parameters are defined below:

parameter	preset	description
NLOCI	2	Number of root locus gains entered in array KGAIN (max=25)
KGAIN		Array of values used for computing root locus gains
	.5	KGAIN(1)=first user-specified root locus gain
	2.	KGAIN(2)=second user-specified root locus gain
		KGAIN(NLOCI)=last root locus gain (Gains computed and used only if they are between KGAIN(1) and KGAIN(NLOCI))
KFLG	0	.EQ.0 to increment gain by multiplying by KDELT .NE.0 to increment gain by adding by KDELT
KDELT	1.E4	Value for changing gains (preset to large value so that no additional gains are computed by the program)
ITLOC	50	Max. no. of different gains computed for root locus
GRAFP	1	.NE.0 for printer (low resolution) plot
FILM	0	.NE.0 for hardcopy (high resolution) plot
RLXMN	0	Minimum x axis for plot
RLXMX	0	Maximum x axis for plot (Auto. scaling of x axis if RLXMN=RLXMX=0)
RLYMN	0	Minimum y axis for plot
RLYMX	0	Maximum y axis for plot (Auto. scaling of y axis if RLYMN=RLYMX=0)

Method

Root locus is computed by evaluating the roots of the polynomial (PN + GAIN*PD) where GAIN is the varied gain and PN and PD are the numerator and denominator polynomials of the transfer function.

SNORM

Identification

SUBROUTINE SNORM - LCAP2 Operator, Normalize S Plane Transfer Function

Purpose

Normalize s plane transfer function using an LCAP2 index. Normalization can be either with respect to the low order non-zero coefficient or the high order coefficient of the denominator.

Usage

```
CALL SNORM(I)
```

I input - Index of the s plane transfer function

1. Normalization parameters are in COMMON/HEADDB/. They are to be set before SNORM is called. These parameters are defined below:

parameter	preset	description
KNORM	1.	Value used for normalizing the transfer function
NRMFG	0	If .EQ.0, the low order non-zero coefficient of the denominator is set equal to the value of KNORM and all other coefficients are normalized to this value. If KNORM=1., the low order non-zero coefficient of the numerator is the Bode gain. If .NE.0, the high order coefficient of the denominator is set equal to the value of KNORM and all other coefficients are normalized to this value. If KNORM=1., the high order coefficient of the numerator is the root locus gain.

Restrictions

KNORM cannot be zero.

SPADD

Identification

SUBROUTINE SPADD - LCAP2 Operator, S Plane Transfer Function Add

Purpose

Add two s plane transfer functions using LCAP2 indices.

Usage

```
CALL SPADD(I,J,K)
```

I input - Index of resultant transfer function sum
J input - Index of first transfer function to be added
K input - Index of second transfer function to be added

SPDIV

Identification

SUBROUTINE SPDIV - LCAP2 Operator, S Plane Transfer Function Divide

Purpose

Divide two s plane transfer functions using LCAP2 indices.

Usage

```
CALL SPDIV(I,J,K)
```

I input - Index of resultant transfer function
J input - Index of dividend transfer function
K input - Index of divisor transfer function

SPEQU

Identification

SUBROUTINE SPEQU - LCAP2 Operator, S Plane Equal

Purpose

Equate s plane transfer functions using LCAP2 indices.

Usage

CALL SPEQU(I,J)

I input - Index of resultant transfer function
J input - Index of transfer function to be equated with

SPLDC

Identification

SUBROUTINE SPLDC - LCAP2 Operator, Load Coefficients Into S Plane Transfer FUNCTION

Purpose

Load coefficients into s plane transfer function using an LCAP2 index.

Usage

CALL SPLDC(I)

I input - Index where transfer function is to be stored

1. Transfer function coefficients are entered with polynomial coefficient arrays POLYN and POLYD (LCAP2 format) which are in COMMON/HEADDB/. They are to be set before SPLDC is called.

SPLDR

Identification

SUBROUTINE SPLDR - LCAP2 Operator, Load S Plane Transfer Function In Root Form

Purpose

Load roots into s plane transfer function using an LCAP2 index.

Usage

CALL SPLDR(I)

I input - Index where transfer function is to be stored

1. Transfer function roots are entered with polynomial coefficient arrays ROOTN and ROOTD (LCAP2 format) which are in COMMON/HEADDB/. They are to be set before SPLDR is called.

SPMPY

Identification

SUBROUTINE SPMPY - LCAP2 Operator, S Plane Transfer Function Multiply

Purpose

Multiply two s plane transfer functions using LCAP2 indices.

Usage

```
CALL SPMPY(I,J,K)
```

I input - Index of resultant transfer function product
J input - Index of first transfer function to be multiplied
K input - Index of second transfer function to be multiplied

Method

If only the coefficients of the j-th and k-th transfer functions are available, the product is computed by multiplication of the coefficients. If the roots of the j-th and k-th transfer functions are available, the product is computed by combining the roots. The coefficients of the product are then formed from these roots.

SPPRN

Identification

SUBROUTINE SPPRN - LCAP2 Operator, Print Out S Plane Transfer Function

Purpose

Print out s plane transfer function using an LCAP2 index.

Usage

```
CALL SPPRN(I)
```

I input - Index of transfer function to be printed out

Method

Roots of the transfer function are printed out only if they are defined (previously computed or loaded in). Coefficients of the transfer function are printed out in ascending order.

SPRTS

Identification

SUBROUTINE SPRTS - LCAP2 Operator, Find Roots Of S Plane Transfer Function

Purpose

Find roots of an s plane transfer function using an LCAP2 index.

Usage

```
CALL SPRTS(I)
```

I input - Index of s plane transfer function

Method

Roots of the numerator and denominator are computed by subroutine PROOT.

Restrictions

If the roots of SPTFi were previously computed or loaded in, the program will not recompute the roots from the coefficients. A message to this effect will be printed.

SPSUB

Identification

SUBROUTINE SPSUB - LCAP2 Operator, S Plane Transfer Function Subtract

Purpose

Subtract two s plane transfer functions using LCAP2 indices.

Usage

```
CALL SPSUB(I,J,K)
```

```
I  input  - Index of resultant transfer function difference
J  input  - Index of first transfer function (minuend)
K  input  - Index of second transfer function (subtrahend)
```

STIME

Identification

SUBROUTINE STIME - LCAP2 Operator, Inverse Laplace Transform And Time Response

Purpose

Compute inverse Laplace transform and the time response using an LCAP2 index.

Usage

```
CALL STIME(I)
```

I input - Index of s plane transfer function

1. Time response parameters are in COMMON/HEADDB/. They are to be set before STIME is called. These parameters are defined below:

parameter	preset	description
TSTEP	1	.NE.0 for step response; .EQ.0 for impulse response
TMAGN	1.	Magnitude of input for time response
TZERO	0	Start time for evaluating time response
TEND	1.	End time for evaluating time response
TDELT	1.	Delta time for evaluating time response
TMAGN	1.	Magnitude of input for time response
GRAFP	1	.NE.0 for printer (low resolution) plot
FILM	0	.NE.0 for hardcopy (high resolution) plot
TXMIN	0	Minimum x axis for plot
TXMAX	0	Maximum x axis for plot (Auto. scaling of x axis if TXMIN=TXMAX)
TYMIN	0	Minimum y axis for plot
TYMAX	0	Mamimum y axis for plot (Auto. scaling of y axis if TYMIN=TYMAX)

Method

The partial fraction expansion of the s plane transfer function times (1/s), if the input is a step function, is first computed. By utilizing the inverse Laplace transform, the analytical solution is computed and printed out. This analytical solution is then evaluated over the range of time values specified.

Restrictions

Due to the algorithm used to implement the partial fraction expansion, the following restrictions on the form of the s plane transfer function apply. Multiple poles are not allowed except for those at the origin. The poles at the origin (including the pole due to the 1/s term if the input is a step) must be 5 or

less. Also, the degree of the numerator must not be greater than the number of poles not at the origin.

STORE

Identification

SUBROUTINE STORE - Store Polynomial, Transfer Function and Matrix Data

Purpose

Store data from an LCAP2 batch job for a restart capability. This data can be accessed in a subsequent batch or interactive job by using the RESTORE operator.

Usage

```
CALL STORE(IPRNFG)
```

IPRNFG input - =0 for no printout of data to be stored

1. To identify the data stored, enter alphanumeric information in HEAD(64) through HEAD(70) of COMMON/HEADDB/ before calling STORE. This information will be printed out when this data is restored in a subsequent job.

Method

Data will be saved on file TAPE31. The first record will be alphanumeric information copied from HEAD(64) through HEAD(70) of COMMON/HEADDB/. The other records on this file are described in the more detailed description of subroutine STORE in Ref. 2.

Restrictions

File type for TAPE31 must be declared with 'FILE,TAPE31,RT=I.' At the end of the batch job the user must catalog file TAPE31 on the CDC 835 with '...,ST=PF6)' in the catalog command.

SWMRX

Identification

SUBROUTINE SWMRX - LCAP2 Operator, S to W Plane Multirate Transform

Purpose

Compute multirate (slow input, fast output) s to w plane transform using LCAP2 indices. The zero order hold, if included, is at the slower input sampling rate. The ratio of the output/input sampling rates must be an integer. (note: the w is not the w' defined by the Tustin's bilinear rule)

Usage

```
CALL SWMRX(I,J)
```

I input - Index of w plane transfer function
J input - Index of s plane transfer function

1. SWMRX parameters are in COMMON/HEADDB/. They are to be set before SWMRX is called. These parameters are defined below:

parameter	preset	description
DELAY	0	Time delay, (enter negative value for time advance)
SAMPT	1.	Sampling period of the faster output sampler
NTGER	1	Integer ratio of output/input sampling rates (or, input/output sampling periods)
ZOH	1	.NE.0 for inclusion of zero order hold at the input

Method

Partial fraction expansion of the s plane transfer function (including the 1/s from the zero order hold if there is one) is computed by subroutine RESDU. The w transform (at the faster output sampling rate) of each term of the expansion is then computed. Next, the terms of the expansion are summed and rationalized. Roots of this intermediate transfer function are then found. This result is multiplied by the discrete contribution of the zero order hold (which is at the slower input sampling rate) to yield the desired transform.

Restrictions

The algorithm used for computing the partial fraction expansion requires the following constraints on the s plane transfer function. Multiple poles are not allowed except for those at the origin. The poles at the origin (including the 1/s from the zero order hold if there is one) must be 5 or less. Also, the degree of the numerator must not be greater than the number of poles not at the origin.

SWXFM

Identification

SUBROUTINE SWXFM - LCAP2 Operator, S to W Plane Transform

Purpose

Compute sampled-data transform from s to w plane using LCAP2 indices. (note: this w is not the w' defined by the Tustin's bilinear rule)

Usage

```
CALL SWXFM(I,J)
```

I input - Index of w plane transfer function
J input - Index of s plane transfer function

1. SWXFM parameters are in COMMON/HEADDB/. They are to be set before SWXFM is called. These parameters are defined below:

parameter	preset	description
DELAY	0	Time delay, (enter negative value for time advance)
SAMPT	1.	Sampling period
ZOH	1	.NE.0 for inclusion of zero order hold

Method

Partial fraction expansion of the s plane transfer function (including the 1/s from the zero order hold if there is one) is computed by subroutine RESDU. The w transform of each term of the expansion is then computed. Next, the terms of the expansion are summed and rationalized. Roots of this intermediate transfer function are then found. This result is multiplied by the discrete contribution of the zero order hold to yield the desired transform.

Restrictions

The algorithm used for computing the partial fraction expansion requires the following constraints on the s plane transfer function. Multiple poles are not allowed except for those at the origin. The poles at the origin (including the 1/s from the zero order hold if there is one) must be 5 or less. Also, the degree of the numerator must not be greater than the number of poles not at the origin.

SZMRX

Identification

SUBROUTINE SZMRX - LCAP2 Operator, S to Z Plane Multirate Transform

Purpose

Compute multirate (slow input, fast output) s to z plane transform using LCAP2 indices. The zero order hold, if included, is at the slower input sampling rate. The ratio of the output/input sampling rates must be an integer.

Usage

```
CALL SZMRX(I,J)
```

I input - Index of z plane transfer function
J input - Index of s plane transfer function

1. SZMRX parameters are in COMMON/HEADDB/. They are to be set before SZMRX is called. These parameters are defined below:

parameter	preset	description
DELAY	0	Time delay, (enter negative value for time advance)
SAMPT	1.	Sampling period of the faster output sampler
NTGER	1	Integer ratio of output/input sampling rates (or, input/output sampling periods)
ZOH	1	.NE.0 for inclusion of zero order hold at the input

Method

Partial fraction expansion of the s plane transfer function (including the 1/s from the zero order hold if there is one) is computed by subroutine RESDU. The w transform (at the faster output sampling rate) of each term of the expansion is then computed. Next, the terms of the expansion are summed and rationalized. Roots of this intermediate transfer function are then found. This result is multiplied by the discrete contribution of the zero order hold (which is at the slower input sampling rate) to yield the w plane form of the desired transform. Next, subroutine WZTRANS is called to transform the w plane roots to the z plane. The coefficients of the z plane transfer function are then computed.

Restrictions

The algorithm use for computing the partial fraction expansion requires that the following constraints on the s plane transfer function. Multiple poles are not allowed except for those at the origin. The poles at the origin (including the 1/s from the zero order hold if there is one) must be 5 or less. Also, the degree of the numerator must not be greater than the number of poles not at the origin.

SZXFM

Identification

SUBROUTINE SZXFM - LCAP2 Operator, S To Z Plane Transform

Purpose

Compute sampled-data transform from s to z plane using LCAP2 indices.

Usage

CALL SZXFM(I,J)

I input - Index of z plane transfer function
J input - Index of s plane transfer function

1. SZXFM parameters are in COMMON/HEADDB/. See description for SWXFM.

Method

Partial fraction expansion of s plane transfer function (including the 1/s from the zero order hold if there is one) is computed by subroutine RESDU. Since calculations performed in the w plane are more accurate than in the z plane, the w transform of each term of the partial fraction expansion is computed. Roots of this intermediate transfer function are then found. This result is multiplied by the discrete contribution of the zero order hold to yield the w plane form of the desired transform. Next, subroutine WZTRANS is called to transform the w plane roots to the z plane. The coefficients of the z plane transfer function are then computed.

Restrictions

Same restrictions on s plane transfer functions that apply for SWXFM. For higher order s plane transfer functions, the w plane transform will be more accurate than the z plane transform. To determine differences in numerical accuracies, compute both w and z plane transforms and compare frequency responses.

WELCR

Identification

SUBROUTINE WELCR - LCAP2 Operator, Eliminate Common Roots Of W Plane Transfer Function

Purpose

Eliminate common roots from a w plane transfer function using an LCAP2 index.

Usage

CALL WELCR(I)

I input - Index of w plane transfer function

1. Common root elimination parameters ECRE1 (preset=2.E-4) and ECRE2 (preset=1.E-8) are in COMMON/HEADDB/.

Method

If a numerator root nrt and a denominator root drt are found such that ABS(drt/nrt - (1.,0.)).LT.ECRE1 for nrt.NE.0 or ABS(drt).LT.ECRE2 for nrt.EQ.0, roots nrt and drt are considered to be common roots and will be eliminated from the transfer function.

WFREQ

Identification

SUBROUTINE WFREQ - LCAP2 Operator, W Plane Frequency Response

Purpose

Evaluate w plane frequency response using an LCAP2 index. Automatic frequency mode available to allow program to dynamically choose its own frequency points to yield a smooth plot of the response.

Usage

CALL WFREQ(I)

I input - Index of w plane transfer function

1. Frequency response parameters are in COMMON/HEADDB. They are to be set before WFREQ is called. See description of SFREQ for the complete list of definitions of these parameters. The parameter SAMPT is described below:

parameter	preset	description
SAMPT	1	Sampling period

Method

Same as that described in detail in description of SFREQ.

In the automatic frequency mode (FAUTO=0) the program will avoid using frequency values at or near the half sampling frequency. W plane frequencies (imaginary part) greater than 1000. will not be used.

WLOCI

Identification

SUBROUTINE WLOCI - LCAP2 Operator, W Plane Root Locus

Purpose

Evaluate w plane root locus using an LCAP2 index. Automatic gain selection available to supplement user-selected gains.

Usage

```
CALL WLOCI(I)
```

I input - Index of w plane transfer function to be evaluated

1. Root locus parameters are in COMMON/HEADDB/. They are to be set before WLOCI is called. See description of SLOCI for a complete list of definition of these parameters.

Method

Same as that described in detail in description of SLOCI.

WMRFQ

Identification

SUBROUTINE WMRFQ - LCAP2 Operator, W Plane Multirate Frequency Response

Purpose

Evaluate multirate (fast input, slow output) frequency response of a w plane transfer function using an LCAP2 index.

Usage

```
CALL WMRFQ(I,M)
```

I input - Index of w plane transfer function
M input - Integer ratio of output/input sampling periods

1. The input w plane transfer function is at the faster sampling rate.

2. Frequency response parameters are in COMMON/HEADDB/. See description of SFREQ.

3. The sampling period, SAMPT, is for the slower output sampler.

Method

The frequency response is evaluated by direct application of Sklansky's frequency decomposition. No explicit rational representation of the slower output transform is computed. If an explicit representation of the slower output transfer function is desired, see LCAP2 operator WMRXFM.

WMRXFM

Identification

SUBROUTINE WMRXFM - LCAP2 Operator, Multirate (fast input, slow output sampler) W Transform

Purpose

Compute the output w transform of a fast to slow sampler using LCAP2 indices. This operation will yield a rational transfer function at the slower sampling rate.

Usage

```
CALL WMRXFM(I,J)
```

I input - Index of resultant slower output w plane transfer function
J input - Index of faster input w plane transfer function

1. The integer ratio, output/input sampling periods, NTGER, of COMMON/HEADDB/ must be set before this subroutine is called.

Method

The faster input transfer function is first transformed to the z plane. The output transfer function of a fast to slow rate sampler is then given in the z plane by Sklansky's frequency decomposition method as,

$$\frac{1}{n} \sum_{k=1}^{n-1} G_n^{T/n}\left(z\, e^{j2*pi*k/n}\right)$$

where $G_n^{T/n}(z)$ is the z plane transfer function at the faster sampling rate and n is the integer ratio of the output/input sampling periods. Using the root form representation of the input transform, a rational representation of the slower rate output transfer function is computed. This transfer function is then transformed to the w plane.

WNORM

Identification

SUBROUTINE WNORM - LCAP2 Operator, Normalize W Plane Transfer Function

Purpose

Normalize w plane transfer function using an LCAP2 index. Normalization can be either with respect to the low order non-zero coefficient or the high order coefficient of the denominator.

Usage

```
CALL WNORM(I)
```

I input - Index of the w plane transfer function

1. Normalization parameters are in COMMON/HEADDB/. They are to be set before WNORM is called. These parameters are defined below:

parameter	preset	description
KNORM	1.	Value used for normalizing the transfer function
NRMFG	0	If .EQ.0, the low order non-zero coefficient of the denominator is set equal to the value of KNORM and all other coefficients are normalized to this value. If KNORM=1., the low order non-zero coefficient of the numerator is the Bode gain. If .NE.0, the high order coefficient of the denominator is set equal to the value of KNORM and all other coefficients are normalized to this value. If KNORM=1., the high order coefficient of the numerator is the root locus gain.

Restrictions

KNORM cannot be zero.

WPADD

Identification

SUBROUTINE WPADD - LCAP2 Operator, W Plane Transfer Function Add

Purpose

Add two w plane transfer functions using LCAP2 indices.

Usage

```
CALL WPADD(I,J,K)
```

I input - Index of resultant transfer function sum
J input - Index of first transfer function to be added
K input - Index of second transfer function to be added

WPDIV

Identification

SUBROUTINE WPDIV - LCAP2 Operator, W Plane Transfer Function Divide

Purpose

Divide two w plane transfer functions using LCAP2 indices.

Usage

```
CALL WPDIV(I,J,K)
```

I input - Index of resultant transfer function
J input - Index of dividend transfer function
K input - Index of divisor transfer function

WPEQU

Identification

SUBROUTINE WPEQU - LCAP2 Operator, W Plane Equal

Purpose

Equate w plane transfer functions using LCAP2 indices.

Usage

CALL WPEQU(I,J)

I input - Index of resultant transfer function
J input - Index of transfer function to be equated with

WPLDC

Identification

SUBROUTINE WPLDC - LCAP2 Operator, Load Coefficients Into W Plane Transfer Function

Purpose

Load coefficients into w plane transfer function using an LCAP2 index.

Usage

CALL WPLDC(I)

I input - Index where transfer function is to be stored

1. Transfer function coefficients are entered with polynomial coefficient arrays POLYN and POLYD (LCAP2 format) which are in COMMON/HEADDB/. They are to be set before WPLDC is called.

WPLDR

Identification

SUBROUTINE WPLDR - LCAP2 Operator, Load W Plane Transfer Function In Root Form

Purpose

Load roots into w plane transfer function using an LCAP2 index.

Usage

```
CALL WPLDR(I)
```

I input - Index where transfer function is to be stored

1. Transfer function roots are entered with polynomial coefficient arrays ROOTN and ROOTD (LCAP2 format) which are in COMMON/HEADDB/. They are to be set before WPLDR is called.

WPMPY

Identification

SUBROUTINE WPMPY - LCAP2 Operator, W Plane Transfer Function Multiply

Purpose

Multiply two w plane transfer functions using LCAP2 indices.

Usage

```
CALL WPMPY(I,J,K)
```

I input - Index of resultant transfer function product
J input - Index of first transfer function to be multiplied
K input - Index of second transfer function to be multiplied

Method

If only the coefficients of the j-th and k-th transfer functions are available, the product is computed by multiplication of the coefficients. If the roots of the j-th and k-th transfer functions are available, the product is computed by combining the roots. The coefficients of the product are then formed from these roots.

WPPRN

Identification

SUBROUTINE WPPRN - LCAP2 Operator, Print Out W Plane Transfer Function

Purpose

Print out w plane transfer function using an LCAP2 index.

Usage

```
CALL WPPRN(I)
```

I input - Index of transfer function to be printed out

Method

Roots of the transfer function are printed out only if they are defined (previously computed or loaded in). Coefficients of the transfer function are printed out in ascending order.

WPRTS

Identification

SUBROUTINE WPRTS - LCAP2 Operator, Find Roots Of W Plane Transfer Function

Purpose

Find roots of a w plane transfer function using an LCAP2 index.

Usage

```
CALL WPRTS(I)
```

I input - Index of w plane transfer function

Method

Roots of the numerator and denominator are computed by subroutine PROOT.

Restrictions

If the roots of WPTFi were previously computed or loaded in, the program will not recompute the roots from the coefficients. A message to this effect will be printed.

WPSUB

Identification

SUBROUTINE WPSUB - LCAP2 Operator, W Plane Transfer Function Subtract

Purpose

Subtract two w plane transfer functions using LCAP2 indices.

Usage

```
CALL WPSUB(I,J,K)
```

```
I  input  - Index of resultant transfer function difference
J  input  - Index of first transfer function (minuend)
K  input  - Index of second transfer function (subtrahend)
```

WSXFM

Identification

SUBROUTINE WSXFM - LCAP2 Operator, Transform W Plane Roots Into S Plane

Purpose

Transform w plane roots into "equivalent" s plane roots using an LCAP2 index. The transformation of the w plane roots to the s plane is not unique. The "equivalent" s plane roots are provided solely to aid the analyst in identifying and correlating w plane roots. The computed s plane roots are not saved.

Usage

```
CALL WSXFM(I)
```

I input - Index of w plane transfer function

1. Sampling period, SAMPT, of COMMON/HEADDB/ must be set before calling this subroutine.

Method

Transformation of the roots from w to the s plane is defined by

```
s = ln( (1+w)/(1-w) ) / SAMPT
```

When w = -1.0 or +1.0 the "equivalent" s plane root is undefined. If ABS(1.-w) is less than 1.R-5, the equivalent root is printed out as 999999.99. If ABS(w+1.) is less than 1.E-5, the equivalent root is printed out as -999999.99.

WZXFM

Identification

SUBROUTINE WZXFM - LCAP2 Operator, W to Z Plane Transformation

Purpose

Compute w to z plane bilinear transformation using an LCAP2 index. (note: the w is not the w' defined by the Tustin's bilinear rule)

Usage

CALL WZXFM(I,J)

I input - Index of computed z plane transfer function
J input - Index of w plane transfer function to be transformed.

1. The sampling period, SAMPT, of COMMON/HEADDB/ must be set before calling this subroutine.

Method

Bilinear transformation is implemented by transformation of the w plane roots.

ZELCR

Identification

SUBROUTINE ZELCR - LCAP2 Operator, Eliminate Common Roots Of Z Plane Transfer Function

Purpose

Eliminate common roots from a z plane transfer function using an LCAP2 index.

Usage

CALL ZELCR(I)

I input - Index of z plane transfer function

1. Common root elimination parameters ECRE1 (preset=2.E-4) and ECRE2 (preset=1.E-8) are in COMMON/HEADDB/.

Method

If a numerator root nrt and a denominator root drt are found such that ABS(drt/nrt - (1.,0.)).LT.ECRE1 for nrt.NE.0 or ABS(drt).LT.ECRE2 for nrt.EQ.0, roots nrt and drt are considered to be common roots and will be eliminated from the transfer function.

ZFREQ

Identification

SUBROUTINE ZFREQ - LCAP2 Operator, Z Plane Frequency Response

Purpose

Evaluate z plane frequency response using an LCAP2 index. Automatic frequency mode available to allow program to dynamically choose its own frequency points to yield a smooth plot of the response.

Usage

CALL ZFREQ(I)

I input - Index of z plane transfer function

1. Frequency response parameters are in COMMON/HEADDB. They are to be set before ZFREQ is called. See description of SFREQ for the complete list of definitions of these parameters. The parameter SAMPT is described below:

parameter	preset	description
SAMPT	1	Sampling period

Method

Same as that described in detail in description of SFREQ.

ZLOCI

Identification

SUBROUTINE ZLOCI - LCAP2 Operator, Z Plane Root Locus

Purpose

Evaluate z plane root locus using an LCAP2 index. Automatic gain selection available to supplement user-selected gains.

Usage

```
CALL ZLOCI(I)
```

I input - Index of z plane transfer function to be evaluated

1. Root locus parameters are in COMMON/HEADDB/. They are to be set before ZLOCI is called. See description of SLOCI for a complete list of definitions of these parameters.

Method

Same as that described in detail in description of SLOCI.

ZMRFQ

Identification

SUBROUTINE ZMRFQ - LCAP2 Operator, Z Plane Multirate Frequency Response

Purpose

Evaluate multirate (fast input, slow output) frequency response of a z plane transfer function using an LCAP2 index.

Usage

```
CALL ZMRFQ(I,M)
```

I input - Index of z plane transfer function
M input - Integer ratio of output/input sampling periods

1. The input z plane transfer function is at the faster sampling rate.

2. Frequency response parameters are in COMMON/HEADDB/. See description of SFREQ.

3. The sampling period, SAMPT, is for the slower output sampler.

Method

The frequency response is evaluated by direct application of Sklansky's frequency decomposition. No explicit rational representation of the slower output transform is computed. If an explicit representation of the slower output transfer function is desired, see LCAP2 operator ZMRXFM.

ZMRXFM

Identification

SUBROUTINE ZMRXFM - LCAP2 Operator, Multirate (fast input, slow output sampler) Z Transform

Purpose

Compute the output z transform of a fast to slow sampler using LCAP2 indices. This operation will yield a rational transfer function at the slower sampling rate.

Usage

```
CALL ZMRXFM(I,J)
```

I input - Index of resultant slower output z transfer function
J input - Index of faster input z transfer function

1. The integer ratio, output/input sampling periods, NTGER, of COMMON/HEADDB/ must be set before this subroutine is called.

Method

The output transform of a fast to slow rate sampler is given by Sklansky's frequency decomposition method as

$$\frac{1}{n}\sum_{k=1}^{n-1} G^{T/n}\left(z_n e^{j2*pi*k/n}\right)$$

where $G^{T/n}(z_n)$ is the z plane transfer function at the faster sampling rate and n is the integer ratio of the output/input sampling periods. Using the root form representation of the input transform, a rational representation of the slower rate output transform is computed.

ZNORM

Identification

SUBROUTINE ZNORM - LCAP2 Operator, Normalize Z Plane Transfer Function

Purpose

Normalize z plane transfer function using an LCAP2 index. Normalization can be either with respect to the low order non-zero coefficient or the high order coefficient of the denominator.

Usage

CALL ZNORM(I)

I input - Index of the z plane transfer function

1. Normalization parameters are in COMMON/HEADDB/. They are to be set before ZNORM is called. These parameters are defined below:

parameter	preset	description
KNORM	1.	Value used for normalizing the transfer function
NRMFG	0	If .EQ.0, the low order non-zero coefficient of the denominator is set equal to the value of KNORM and all other coefficients are normalized to this value. If .NE.0, the high order coefficient of the denominator is set equal to the value of KNORM and all other coefficients are normalized to this value.

Restrictions

KNORM cannot be zero.

ZPADD

Identification

SUBROUTINE ZPADD - LCAP2 Operator, Z Plane Transfer Function Add

Purpose

Add two z plane transfer functions using LCAP2 indices.

Usage

```
CALL ZPADD(I,J,K)
```

```
I  input  - Index of resultant transfer function sum
J  input  - Index of first transfer function to be added
K  input  - Index of second transfer function to be added
```

ZPDIV

Identification

SUBROUTINE ZPDIV - LCAP2 Operator, Z Plane Transfer Function Divide

Purpose

Divide two z plane transfer functions using LCAP2 indices.

Usage

```
CALL ZPDIV(I,J,K)
```

```
I  input  - Index of resultant transfer function
J  input  - Index of dividend transfer function
K  input  - Index of divisor transfer function
```

ZPEQU

Identification

SUBROUTINE ZPEQU - LCAP2 Operator, Z Plane Equal

Purpose

Equate z plane transfer functions using LCAP2 indices.

Usage

CALL ZPEQU(I,J)

I input - Index of resultant transfer function
J input - Index of transfer function to be equated with

ZPLDC

Identification

SUBROUTINE ZPLDC - LCAP2 Operator, Load Coefficients Into Z Plane Transfer Function

Purpose

Load coefficients into z plane transfer function using an LCAP2 index.

Usage

CALL ZPLDC(I)

I input - Index where transfer function is to be stored

1. Transfer function coefficients are entered with polynomial coefficient arrays POLYN and POLYD (LCAP2 format) which are in COMMON/HEADDB/. They are to be set before ZPLDC is called.

ZPLDR

Identification

SUBROUTINE ZPLDR - LCAP2 Operator, Load Z Plane Transfer Function In Root Form

Purpose

Load roots into z plane transfer function using an LCAP2 index.

Usage

```
CALL ZPLDR(I)
```

I input - Index where transfer function is to be stored

1. Transfer function roots are entered with polynomial coefficient arrays ROOTN and ROOTD (LCAP2 format) which are in COMMON/HEADDB/. They are to be set before ZPLDR is called.

ZPMPY

Identification

SUBROUTINE ZPMPY - LCAP2 Operator, Z Plane Transfer Function Multiply

Purpose

Multiply two z plane transfer functions using LCAP2 indices.

Usage

```
CALL ZPMPY(I,J,K)
```

I input - Index of resultant transfer function product
J input - Index of first transfer function to be multiplied
K input - Index of second transfer function to be multiplied

Method

If only the coefficients of the j-th and k-th transfer functions are available, the product is computed by multiplication of the coefficients. If the roots of the j-th and k-th transfer functions are available, the product is computed by combining the roots. The coefficients of the product are then formed from these roots.

ZPPRN

Identification

SUBROUTINE ZPPRN - LCAP2 Operator, Print Out Z Plane Transfer Function

Purpose

Print out z plane transfer function using an LCAP2 index.

Usage

```
CALL ZPPRN(I)
```

I input - Index of transfer function to be printed out

Method

Roots of the transfer function are printed out only if they are defined (previously computed or loaded in). Coefficients of the transfer function are printed out in ascending order.

ZPRTS

Identification

SUBROUTINE ZPRTS - LCAP2 Operator, Find Roots Of Z Plane Transfer Function

Purpose

Find roots of a z plane transfer function using an LCAP2 index.

Usage

```
CALL ZPRTS(I)
```

I input - Index of z plane transfer function

Method

Roots of the numerator and denominator are computed by subroutine PROOT.

The code for this routine is in subroutine SPRTS.

Restrictions

If the roots of ZPTFi were previously computed or loaded in, the program will not recompute the roots from the coefficients. A message to this effect will be printed.

ZPSUB

Identification

SUBROUTINE ZPSUB - LCAP2 Operator, Z Plane Transfer Function Subtract

Purpose

Subtract two z plane transfer functions using LCAP2 indices.

Usage

```
CALL ZPSUB(I,J,K)
```

I input - Index of resultant transfer function difference
J input - Index of first transfer function (minuend)
K input - Index of second transfer function (subtrahend)

ZSXFM

Identification

SUBROUTINE ZSXFM - LCAP2 Operator, Transform Z Plane Roots Into S Plane

Purpose

Transform z plane roots into "equivalent" s plane roots using an LCAP2 index. The transformation of the z plane roots to the s plane is not unique. The "equivalent" s plane roots are provided solely to aid the analyst in identifying and correlating z plane roots. The computed s plane roots are not saved.

Usage

```
CALL ZSXFM(I)
```

I input - Index of z plane transfer function

1. Sampling period, SAMPT, of COMMON/HEADDB/ must be set before calling this subroutine.

Method

Transformation of the roots from z to the s plane is defined by

$$s = \ln(z) / SAMPT$$

ZTIME

Identification

SUBROUTINE ZTIME - LCAP2 Operator, Inverse Z Transform And Time Response

Purpose

Compute inverse z transform and the time response using an LCAP2 index.

Usage

CALL ZTIME(I)

I input - Index of z plane transfer function

1. Time response parameters are in COMMON/HEADDB/. They are to be set before ZTIME is called. These parameters are defined below:

parameter	preset	description
TSTEP	1	.NE.0 for step response; .EQ.0 for impulse response
TMAGN	1.	Magnitude of input for time response
TEND	1.	End time for evaluating time response
TMAGN	1.	Magnitude of input for time response
SAMPT	1.	Sampling period
GRAFP	1	.NE.0 for printer (low resolution) plot
FILM	0	.NE.0 for hardcopy (high resolution) plot
TXMIN	0	Minimum x axis for plot
TXMAX	0	Maximum x axis for plot
		(Auto. scaling of x axis if TXMIN=TXMAX)
TYMIN	0	Minimum y axis for plot
TYMAX	0	Mamimum y axis for plot
		(Auto. scaling of y axis if TYMIN=TYMAX)

Method

The inverse z transform is computed by the power series (long division) method. While this method of computing the time response is inherently less accurate than the partial fraction method, results for typical transfer functions are excellent. To provide a measure of the accuracy of the response, the results are computed in double precision and compared with single precision results.

ZVCNG

Identification

SUBROUTINE ZVCNG - LCAP2 Operator, Z to (Z**n) Transformation

Purpose

Compute transformation of z plane transfer function to a faster z variable by replacement of variables.

Usage

```
CALL ZVCNG(I,J)
```

I input - Index of resultant z plane transfer function expressed in terms of the faster z variable

J input - Index of z plane transfer function to be operated upon

1. The integer ratio of faster/slower sampling rate, NTGER, of COMMON/HEADDB/ must be set before ZVCNG is called.

Method

The z variable of the slower sampled transfer function is replaced by z**n and stored into the faster sampled transfer function. This subroutine calls ZVCHNG1.

Restrictions

Since the format used to represent transfer function arrays in LCAP2 limits the degree of the polynomials to less that 50, the degree of the j-th z plane transfer function times NTGER must be less than 50.

ZWXFM

Identification

SUBROUTINE ZWXFM - LCAP2 Operator, Z to W Plane Transformation

Purpose

Compute z to w plane bilinear transformation using an LCAP2 index. (note: the w is not the w' defined by the Tustin's bilinear rule)

Usage

```
CALL ZWXFM(I,J)
```

I input - Index of computed w plane transfer function
J input - Index of z plane transfer function to be transformed.

1. The sampling period, SAMPT, of COMMON/HEADDB/ must be set before calling this subroutine.

Method

Bilinear transformation is implemented by transformation of the z plane roots.

APPENDIX B - DEFINITION OF LCAP2 PARAMETERS

All of the following parameters are initialized by subroutines INITO and MINITO which should be called as the first and second executable statements in the main program.

Param.	Preset		Description
CONTP	0	-	Film plot continuation flag (cannot be used for Bode plots) .EQ.0 - Single plot .EQ.1 - First curve of a plot .EQ.2 - Continuation of a plot .EQ.3 - Final curve of a plot
CYCLE	0	-	.EQ.0 for automatic selection of 2 or 3 cycle scal e for BODE plots. (1 cycle is not available)
DBMAX	0	-	Max. db for plotting frequency response
DBMIN	0	-	Min. db for plotting frequency response (auto. scaling if DBMAX.EQ.DBMIN)
DEGMN	-360.	-	Min. defined phase in frequency response (phase defined from DEGMN to DEGMN+360.)
DELAY	0	-	Delay time for sampled-data transform - (sec.)
ECRE1	2.E-4	-	Tolerance for common root in common root elimination subroutine CRELIM
ECRE2	1.E-8	-	Tolerance for zero root in subroutine CRELIM
EDB1	1.	-	Min. delta db in freq. response for plotting
EDB2	2.	-	Min. delta db in freq. response for print out
EDEG1	4.	-	Min. delta deg. in freq. response for plotting
EDEG2	10.	-	Min. delta deg. in freq. response for print out
EPAD1	1.E-10	-	Tolerance for negligible higher order coefficients in subroutine ADDP
EP1	1.E-8	-	An input quantity for subroutine MULE
EP2	1.E-10	-	An input quantity for subroutine MULE
EP3	.01	-	An input quantity for subroutine MULE
EP4	1.E-8	-	An input quantity for subroutine MULE
EP5	1.E-5	-	An input quantity for subroutine MULE
ERCNJ	1.E-4	-	Tolerance to determine if complex root pairs are conjugates
ERCX	1.E-4	-	Roots are considered to be complex if imag. part .GT. ERCX
ERCZ	1.E-5	-	Roots are considered to be zero if CABS.LT.ERCZ
FAUTO	1	-	.NE.0 for automatic frequency mode. Uses NOMEG and OMEGA arrays. .EQ.0 for user supplied frequency points. Uses FREQ1, FREQ2, ..., FREQ5 arrays.
FBODE	1	-	.NE. 0 for Bode plots with frequency response
FDLAY	0	-	Time delay for s-plane frequency response

FILM	0	- .NE.0 for hardcopy (high resolution) plots
FNICO	0	- .NE.0 for Nichols plot with frequency response
FNYQS	0	- .NE.0 for Nyquist plot with frequency response
FREQ1(1)	1.	- Starting frequency point for first segment of user specified values,
FREQ1(2)	10.	- End frequency point for second segment of user specified values,
FREQ3(1)	1.	- Delta frequency for third segment of user specified values,
FREQk(1)	0	- Starting frequency point for k-th segment, *
FREQk(2)	0	- End frequency point for k-th segment, *
FREQk(3)	0	- Delta frequency for k-th segment, *
(k=2,5)		(* - only if FAUTO.EQ.0)
FXYDL	.5	- Nyquist plot scale in units per inch (Auto. scaling if FXYDL=0)
FXYMN	-2.5	- Nyquist plot parameter - minimum real and imag. value plotted
GRAFP	1	- .NE.0 for printer (low resolution) plots
ITLOC	50	- Max. no. of different gains computed in root locus
KDELT	1.E+4	- Value for changing root locus gains (preset to large value so that no additional gains are computed by the program)
KFLG	1	- Flag for computing root locus gains .EQ.0 for computing gains by ratio KDELT .NE.0 for computing gains by increment KDELT
KGAIN		- Array of values used for computing root locus gains
	.5	KGAIN(1)=first user specified root locus gain
	2.	KGAIN(2)=second user specified root locus gain
		.
		.
		KGAIN(NLOCI)=last root locus gain (Gains computed and used only if they are between KGAIN(1) and KGAIN(NLOCI))
KNORM	1	- Gain used for normalizing polynomial or transfer function
MAXIT	80	- Max. no. of iterations allowed per root in subroutine MULE
MDEG	0	- Degree of highest order polynomial in matrix (0-4)
MNDW	.0005	- Min. relative freq. step size in freqy. response when FAUTO.NE.0
MTGER	1	- Integer m for multirate configuration
MXDW	.22	- Max. relative freq. step size in freqy. response when FAUTO.NE.0
MXITF	3000	- Max. no. of iterations in auto. mode of freq. response
MXM	1	- Dimension of matrices (1-30)
M0	0	- Matrix for coefficients of s**0
M1	0	- Matrix for coefficients of s**1
M2	0	- Matrix for coefficients of s**2
M3	0	- Matrix for coefficients of s**3
M4	0	- Matrix for coefficients of s**4

NANOT	4	-	Number of lines of annotations on hardcopy (high resolution electrostatic) plot (0-4)
NLOCI	2	-	Number of root locus gains entered in array KGAIN (max=25)
NOMEG	2	-	Number of frequency points entered in OMEGA array for use in auto. frequency mode (max=20)
NP	2	-	Flag for determining output from subroutine MULE .EQ.0 - print all iterants and BCI output for special procedures .EQ.1 - print only the final iteration of each root .EQ.2 - suppress all internal printing (note - program always print final iteration if max iteration obtained)
NQDB	0	-	.NE.0 for hardcopy Nyquist plot in db
NRMFG	0	-	Polynomial And Transfer Function Normalization Flag If .EQ.0, the low order non-zero coefficient of the denominator is set equal to the value of KNORM and all other coefficients are normalized to this value. For the s plane, if KNORM=1., the low order non-zero coefficient of the numerator is the Bode gain. If .NE.0, the high order coefficient of the denominator is set equal to the value of KNORM and all other coefficients are normalized to this value. For the s plane, if KNORM=1., the high order coefficient of the numerator is the root locus gain.
NTGER	1	-	Integer n for multirate configuration
OMEGA		-	Array of frequency points for auto. frequency mode (units defined by RAD)
	1.		OMEGA(1) = first frequency point used in auto. mode
	10.		OMEGA(2) = second frequency point used in auto. mode . . OMEGA(NOMEG) = last freq. point used in auto. mode
PMARG	0	-	.NE. 0 for plotting phase margin instead of phase for the Nichols plot
POLY	0	-	Array used to input coefficients for polynomials
POLYN	0	-	Array used to input numerator coefficients for transfer functions
POLYD	0	-	Array used to input denominator coefficients for transfer functions
PRNFLG1	1	-	(not used)
PRNFLG2	1	-	.EQ.0 to suppress print out of arguments of an LCAP2 operation
PRNFLG3	1	-	.EQ.0 to suppress print out of LCAP2 operation executed
PRNFLG4	1	-	.EQ.0 to suppress print out of results of an LCAP2 operation
PRNFLG5	1	-	(not used)
PRN1	0	-	.EQ.0 for suppressing print out from CRELIM
PRN2	0	-	.EQ.0 for suppressing print out from FREQS

PRN3	0	-	.EQ.0 for suppressing print out from RCLAS
PRN4	0	-	.EQ.0 for suppressing print out from RESDU
PRN5	0	-	.EQ.0 for suppressing print out from STIME
PRN6	0	-	.EQ.0 for suppressing print out from WTRANS
PRN7	0	-	.EQ.0 for suppressing print out from PROOT and MROOT1
PRN10	0	-	.NE.0 for print out of partial fraction coefficients from subroutine RESDU
RAD	1	-	.NE.0 For frequency in rad/sec .EQ.0 For frequency in Hz
RLFG1	-1	-	Flag for numbering root locus points on the hardcopy (higher resolution) plots = 1 for numbering, = -1 for no numbering
RLXMN	-450.	-	Root locus plot param., min. x scale
RLXMX	50.	-	Root locus plot param., max. x scale (auto. scaling of x axis if RLXMN=RLYMX)
RLYMN	-50.	-	Root locus plot param., min. y scale
RLYMX	450.	-	Root locus plot param., max. y scale (auto. scaling of y axis if RLYMN=RLYMX)
ROOT	0	-	Array used to input roots of polynomials
ROOTN	0	-	Array used to input numerator roots for transfer functions
ROOTD	0	-	Array used to input denominator roots for transfer functions
RTMAX	1.E+7	-	Max. root to be found by subroutine MULE
RZERO	1.E-5	-	Roots .LT. RZERO returned from subroutine MULE are set identically to zero
SAMPT	1	-	Sampling period - (sec.)
SHADE	16	-	Pen intensity for electrostatic plot (2-28)
TDELT	1.	-	Incremental time for s-plane time response
TEND	1.	-	End time for time response calculation
TMAGN	1.	-	Magnitude of input for time response calculation
TSTEP	1	-	.NE.0 for step response when computing time response .EQ.0 for impulse response when computing time response
TXMAX	0	-	Maximum x axis for time response plot
TXMIN	0	-	Minimum x axis for time response plot (auto. scaling of time axis if TXMIN=TXMAX)
TYMAX	0	-	Maximum y axis for time response plot
TYMIN	0	-	Minimum y axis for time response plot (auto. scaling of y axis if TYMIN=TYMAX)
TZEP1	1.	-	Inverse z transform computational error criteria. Percent difference between single and double precision for which time of first occurrence will be printed out
TZERO	0	-	Start time for evaluating s plane time response
TZFLG	0	-	.EQ.0 for step plot in inverse z transform response plot (hardcopy only) .NE.0 for symbol plot in inverse z transform response plot (hardcopy only)
XGAP	50.	-	Plot pen lifts up when x distance exceeds XGAP*(.01) inches
YANOT	9.6	-	Y position for second line of hardcopy plot annotation

			(in.) (range is 0-10)
YGAP	50.	-	Plot pen lifts up when y distance exceeds YGAP*(.01) inches
ZLINE	19.	-	Pen intensity for accenting zero line for hardcopy plots
ZOH	1	-	.NE.0, include zero order hold in computation of sampled-data transform .EQ.0, do not include zero order hold

APPENDIX C - DESCRIPTION OF SOME COMMONLY USED LCAP2 SUBROUTINES

HEADINI,(I=1,5)

Identification

SUBROUTINE HEADINi - Heading Statement For Entering Plot Titles (i=1,5)

Purpose

Simple FORTRAN statement for entering Hollerith data into the plot array HEAD of COMMON/HEADDB/ used for labeling plot titles.

Usage

```
     CALL HEADINi(INDX,WORD)

INDX  input  - Pointer to array HEAD of COMMON/HEADDB/ where array
               WORD will be copied into
WORD  input  - Hollerith data with format 10H... if i=1
                                          20H... if i=2
                                           .        .
                                          50H... if i=5
```

1. First line of plot title is in HEAD(i),i=1,7
   Second line of plot title is in HEAD(i),i=8,14
   Third line of plot title is in HEAD(i),i=15,21
   Fourth line of plot title is in HEAD(i),i=22,28

2. Example: CALL HEADIN2(8,20HTHIS IS AN EXAMPLE) will yield,

   ```
   HEAD(3)=10HTHIS IS AN
   HEAD(4)=10HEXAMPLE
   ```

   which will appear as the second line of the plot title.

3. First line of plot title will appear at the top of the plot. Second line will begin a YANOT units from the bottom of page (full scale defined from 0-10 units). YANOT (preset=9.6) is in COMMON/HEADDB/.

REMARKI,(I=1,5)

Identification

SUBROUTINE REMARKi - Print Out Remarks, (i=1,5)

Purpose

Single FORTRAN statement for printing out Hollerith data.

Usage

```
CALL REMARKi(A)
```

```
A  input  - Hollerith data with format 10H... if i=1
                                       20H... if i=2
                                       30H... if i=3
                                       40H... if i=4
                                       50H... if i=5
```

1. Example: CALL REMARK2(20HTHIS IS AN EXAMPLE) will print out,

   THIS IS AN EXAMPLE

SFAUX

Identification

COMPLEX FUNCTION SFAUX - Evaluate S Plane Transfer Function Coefficient Array

Purpose

Evaluate s plane transfer function coefficient array (LCAP2 format) for use in computing the frequency response. This complex function can be be used by subroutine FREQS1 or FREQS2 to evaluate the transfer function specified by its first argument. It can also be used by user-supplied subroutines similar to SFAUX1.

Usage

SFAUX(TFC)

TFC input - Transfer function coefficient array (LCAP2 format)
SFAUX output - Complex value of response

1. Independent s plane frequency to be used in evaluating the response is determined in subroutine FREQS1 or FREQS2.

SFAUX1

Identification

COMPLEX FUNCTION SFAUX1 - Evaluate S Plane Transfer Function Coefficient Array

Purpose

This complex function is similar to SFAUX except that it is written so that it can be easily modified by the user to allow creation of a user-defined s plane transfer function.

Usage

SFAUX1(TFC)

TFC	input	- Transfer function coefficient array (LCAP2 format)
SFAUX1	output	- Complex value of response

Method

This complex function has only one line of code

```
SFAUX1=SFAUX(TFC)
```

so that it will yield the same results as SFAUX.

To create a user defined s plane transfer function, a different value is returned for SFAUX1. For example, if the function is

SPTF2 + SPTF4/2.

the user would change the FORTRAN code to

```
SFAUX1=SFAUX(SPTF2) + SFAUX(SPTF4)/2.
```

Restrictions

The argument TFC must be an array in memory. Since only the first five s plane transfer functions are in COMMON/SCMBLKB/ and all others are on a disk file, only SPTF1, SPTF2, SPTF3, SPTF4 and SPTF5 can be used to create a user-defined s plane transfer function. However, the user can define additional transfer function coefficient arrays in a separate labeled common block to be accessible by SFAUX. Subroutine FETSTF can be used to copy transfer functions from the disk file to the transfer function in this common block.

The code for this routine is included so that Example 16, which uses this routine, can be better understood. The code for is routine is

```
---------------------------------------------------------------------------
      COMPLEX FUNCTION SFAUX1(TFC)
C     THIS FUNCTION WILL YIELD RESULTS WHICH ARE IDENTICAL TO
C     FUNCTION SFAUX IF STATEMENT 100 BELOW IS NOT CHANGED.
C
C     THE INTENT OF THIS FUNCTION IS TO PROVIDE THE LCAP2 USER WITH
C     A ROUTINE WHICH CAN BE EASILY MODIFIED TO ALLOW CREATION OF A USER
C     DEFINED S-PLANE FUNCTION TO BE USED WITH SUBROUTINE FREQS. THE
C     PROCEDURE IS TO REPLACE STATEMENT 100 WITH A USER DEFINED
C     FUNCTION.
C
C     **************************************************************
C     * EXAMPLE - IF THE USER DEFINED FUNCTION IS TO BE             *
C     *                                                             *
C     *           SPTF2 + SPTF4/2.                                  *
C     *                                                             *
C     *           REPLACE STATEMENT 100 BY                          *
C     *                                                             *
C     *       100 SFAUX1 = SFAUX(SPTF2) + SFAUX(SPTF4)/2.           *
C     **************************************************************
C     *************************** NOTE *****************************
C
C     ONLY THE 5 S-PLANE TRANSFER FUNCTIONS SPTF1,SPTF2,SPTF3,SPTF4,SPTF5
C     CAN BE USED DIRECTLY WITH FUNCTION SFAUX. ALL OTHER STORED S-PLANE
C     TRANSFER FUNCTIONS ARE ON DISK STORAGE AND THEREFORE NOT DIRECTLY
C     ACCESSIBLE WITH FUNCTION SFAUX.
C     IN ORDER TO USE ANY OF THESE OTHER S-PLANE TRANSFER FUNCTIONS,
C     THE USER MUST (1) DECLARE AN SCM COMMON BLOCK IN THE MAIN PROGRAM
C     AND IN SFAUX1, AND (2) TRANSFER THE DESIRED TRANSFER FUNCTIONS
C     INTO THIS COMMON BLOCK. SEE E.A. LEE FOR DETAILS.
C
      COMMON/FRQBLK/U,X,TWOPI,MMTGER
      COMPLEX X
      COMMON/HEADDB/HEAD(70),DB(900)
      COMMON/SCMBLK/XTFS(1520),XTFW(1520),XTFZ(1520),XPY(760)
      DIMENSION SPTF1(102),SPTF2(102),SPTF3(102),SPTF4(102),SPTF5(102)
      EQUIVALENCE (XTFS(1),SPTF1(1)),(XTFS(305),SPTF2(1))
     +,(XTFS(609),SPTF3(1)),(XTFS(913),SPTF4(1)),(XTFS(1217),SPTF5(1))
      DIMENSION WPTF1(102),WPTF2(102),WPTF3(102),WPTF4(102),WPTF5(102)
      EQUIVALENCE (XTFW(1),WPTF1(1)),(XTFW(305),WPTF2(1))
     +,(XTFW(609),WPTF3(1)),(XTFW(913),WPTF4(1)),(XTFW(1217),WPTF5(1))
      DIMENSION ZPTF1(102),ZPTF2(102),ZPTF3(102),ZPTF4(102),ZPTF5(102)
      EQUIVALENCE (XTFZ(1),ZPTF1(1)),(XTFZ(305),ZPTF2(1))
     +,(XTFZ(609),ZPTF3(1)),(XTFZ(913),ZPTF4(1)),(XTFZ(1217),ZPTF5(1))
      COMPLEX SFAUX
  100 SFAUX1=SFAUX(TFC)
      RETURN
      END
```

SFREQY

Identification

SUBROUTINE SFREQY - Evaluate Frequency Response Of An S Plane Transfer Function Coefficient Array

Purpose

Evaluate frequency response of an s plane transfer function coefficient array. User supplies name of the array.

Usage

```
CALL SFREQY(TFC)
```

TFC input - Transfer function coefficient array (LCAP2 format)

1. Frequency response parameters are in COMMON/HEADDB/. See description of subroutine SFREQ in Appendix A for definitions.

Restrictions

If LCAP2 defined transfer function coefficient arrays are to be used, only the first five transfer functions for each plane are available, since the others are on disk files. However, a user common block can be defined so that these other transfer functions can be first transferred from disk file to memory with subroutine FETSTF so that SFREQY can be used.

WFAUX

Identification

COMPLEX FUNCTION WFAUX - Evaluate W Plane Transfer Function Coefficient Array

Purpose

Evaluate w plane transfer function coefficient array (LCAP2 format) for use in computing the frequency response. This complex function can be used by subroutine FREQW1 or FREQW2 to evaluate the transfer function specified by its first argument. It can also be used by user-supplied subroutines similar to WFAUX1.

This subroutine can also evaluate the multirate (fast input, slow output) response of the transfer function.

Usage

WFAUX(TFC)

TFC   input - Transfer function coefficient array (LCAP2 format)
WFAUX output - Complex value of response

1. Independent w plane frequency used in evaluation of the response is computed by the program using real frequency X of COMMON/FRQBLK/ and sampling period SAMPT of COMMON/HEADDB/.

2. If MMTGER of COMMON/FRQBLK/ is .GT.0, the multirate response is computed by using Sklansky's frequency decomposition method. MMTGER is the ratio of the (output/input) sampling periods and SAMPT is the sampling period of the faster input sampler.

WFAUX1

Identification

COMPLEX FUNCTION WFAUX1 - Evaluate W Plane Transfer Function Coefficient Array

Purpose

This complex function is similar to WFAUX except that it is written so that it can be easily modified by the user to allow creation of a user-defined w plane transfer function.

Usage

```
WFAUX1(TFC)
```

```
TFC    input  - Transfer function coefficient array (LCAP2 format)
WFAUX1 output - Complex value of the response
```

Method

This complex function has only one line of code:

```
WFAUX1=WFAUX(TFC)
```

so that it will yield the same results as WFAUX.

To create a user defined w plane transfer function, a different value is returned for WFAUX1. For example, if the function is:

```
WPTF2 + WPTF4/2.
```

the user would change the FORTRAN code to

```
WFAUX1=WFAUX(WPTF2) + WFAUX(WPTF4)/2.
```

WFREQY

Identification

SUBROUTINE WFREQY - Evaluate Frequency Response Of A W Plane Transfer Function Coefficient Array

Purpose

Evaluate frequency response of a w plane transfer function coefficient array. User supplies name of the array.

Usage

```
CALL WFREQY(TFC)
```

TFC input - Transfer function coefficient array (LCAP2 format)

1. Frequency response parameters are in COMMON/HEADDB/. See description of subroutine SFREQ for definition.

Restrictions

If LCAP2 defined transfer function coefficient arrays are to be used, only the first five transfer functions for each plane are available, since the others are on disk files. However, a user common block can be defined so that these other transfer functions can be first transferred from disk file to memory so that WFREQY can be used.

ZFAUX

Identification

COMPLEX FUNCTION ZFAUX - Evaluate Z Plane Transfer Function Coefficient Array

Purpose

Evaluate z plane transfer function coefficient array (LCAP2 format) for use in computing the frequency response. This complex function can be used by subroutine FREQZ1 or FREQZ2 to evaluate the transfer function specified by its first argument. It can also be used by user-supplied subroutines similar to ZFAUX1.

This subroutine can also evaluate the multirate (fast input, slow output) response of the transfer function.

Usage

ZFAUX(TFC)

TFC   input - Transfer function coefficient array (LCAP2 format)
ZFAUX output - Complex value of response

1. Independent z plane frequency used in evaluation of the response is computed by the program using real frequency X of COMMON/FRQBLK/ and sampling period SAMPT of COMMON/HEADDB/.

2. If MMTGER of COMMON/FRQBLK/ is .GT.0, the multirate response is computed by using Sklansky's frequency decomposition method. MMTGER is the ratio of the (output/input) sampling periods and SAMPT is the sampling period of the faster input sampler.

ZFAUX1

Identification

COMPLEX FUNCTION ZFAUX1 - Evaluate Z Plane Transfer Function Coefficient Array

Purpose

This complex function is similar to ZFAUX except that it is written so that it can be easily modified by the user to allow creation of a user-defined z plane transfer function.

Usage

```
ZFAUX1(TFC)
```

TFC input - Transfer function coefficient array (LCAP2 format)
ZFAUX1 output - Complex value of the response

Method

This complex function has only one line of code

```
ZFAUX1=ZFAUX(TFC)
```

so that it will yield the same results as ZFAUX.

To create a user defined z plane transfer function, a different value is returned for ZFAUX1. For example, if the function is

```
ZPTF2 + ZPTF4/2.
```

the user would change the FORTRAN code to

```
ZFAUX1=ZFAUX(ZPTF2) + ZFAUX(ZPTF4)/2.
```

ZFREQY

Identification

SUBROUTINE ZFREQY - Evaluate Frequency Response Of A Z Plane Transfer Function Coefficient Array

Purpose

Evaluate frequency response of a z plane transfer function coefficient array. User supplies name of the array.

Usage

```
CALL ZFREQY(TFC)
```

TFC input - Transfer function coefficient array (LCAP2 format)

1. Frequency response parameters are in COMMON/HEADDB/. See description of subroutine SFREQ for definition.

Restrictions

If LCAP2 defined transfer function coefficient arrays are to be used, only the first five transfer functions for each plane are available, since the others are on disk files. However, a user common block can be defined so that these other transfer functions can be first transferred from disk file to memory so that ZFREQY can be used.

APPENDIX D - NOTES ON LABELING PLOTS

Alphanumeric information for labeling plots are in array HEAD of COMMON/HEADDB/. For hardcopy (high resolution electostatic) plots, up to four lines of annotation are available. For printer plots, only one line of annotation is available. The data in this array is used as follows:

```
HEAD(i), i=1,7     for 1st line, hardcopy and printer plots
HEAD(i), i=8,14    for 2nd line, hardcopy plots only
HEAD(i), i=15,21   for 3rd line, hardcopy plots only
HEAD(i), i=22,28   for 4th line, hardcopy plots only
```

This HEAD array is preset to blanks. Whenever a hardcopy plot is made the contents in the HEAD array will be printed out. The user must enter or change the contents in this array prior to a FORTRAN call (an LCAP2 operator) which produces a plot.

For example, if the first line is to be "EXAMPLE 1 S PLANE FREQUENCY RESPONSE," either one of the following can be used:

Method 1

```
HEAD(1)=10HEXAMPLE 1
HEAD(2)=10HS PLANE FR
HEAD(3)=10HEQUENCY RE
HEAD(4)=10HSPONSE
```

Method 2

```
CALL HEADIN4(1,40HEXAMPLE 1 S PLANE FREQUENCY RESPONSE    )
```

Method 3

```
     ENCODE(40,100,HEAD(1))
 100 FORMAT(40HEXAMPLE 1 S PLANE FREQUENCY RESPONSE    )
```

Method 2 is the simplest to use since only a single line of FORTRAN code is required. This subroutine is of the form HEADINi,(i=1,5) where i designates the multiples of 10 characters to be entered and the first argument of the subroutine designates the starting location of the HEAD array where the alphanumeric information is to be stored. For more details, see description of this subroutine in Appendix C.

Method 3 has the advantage of allowing the user to annotate the plots with data if the ENCODE statement were to include a list. For example, if the second

line of annotation on the hardcopy plot were to be labeled with the values of the variables FLEXW and ZETA, the following FORTRAN statements could be used:

```
      ENCODE(31,200,HEAD(8))FLEXW,ZETA
  200 FORMAT(11HFLEXMODE =,F7.3,8H, ZETA =,F5.3)
```

If the values of FLEXW and ZETA were 14.523 and .707, respectively, HEAD(8) through HEAD(11) would have the following characters in it.

HEAD(8)	HEAD(9)	HEAD(10)	HEAD(11)
FLEX MODE	= 14.523,	ZETA = .70	7

The first line of annotation will always be at the top of the plot. The second, third and fourth lines (hardcopy plots only) normally begin just below the first line. These last three lines, however, can be placed lower on the plot if they should interfere with the plot data. The parameter YANOT (preset to 9.6) of COMMON/HEADDB/, which positions the y coordinate for the second line, can be changed to a smaller value. The range for this parameter is 0-10, which corresponds to the length of the y axis.

For the hardcopy plots the number of lines of annotation to be produced is determined by the parameter ANOTAT of COMMON/HEADDB/, which is preset to 4.

APPENDIX E - PROGRAM AVAILABILITY

The source code for this program is available to agencies supporting DOD projects and studies. The requester, however, should be aware that some non-ANSI FORTRAN code and one assembly language subroutine are utilized. If the program is be be run on a CDC 176 or 7600 computer under the SCOPE 2.1 operating system, no problems should be encountered. If the program is to be run on any other computer, modifications to the program most likely will have to be made. The following facts will be of interest if modifications are to be made:

(a) A FORTRAN version of CXMTX1, which is written in assembly language, is available.

(b) Non-ANSI CDC FORTRAN 4 ENCODE and DECODE statements must be replaced with ANSI standard internal write and read statements. Even if the target computer and operating system supports the ENCODE and DECODE statements modifications might still be necessary since the ENCODE statements used in LCAP2 makes use of the fact that the CDC word length is 60 bits long.

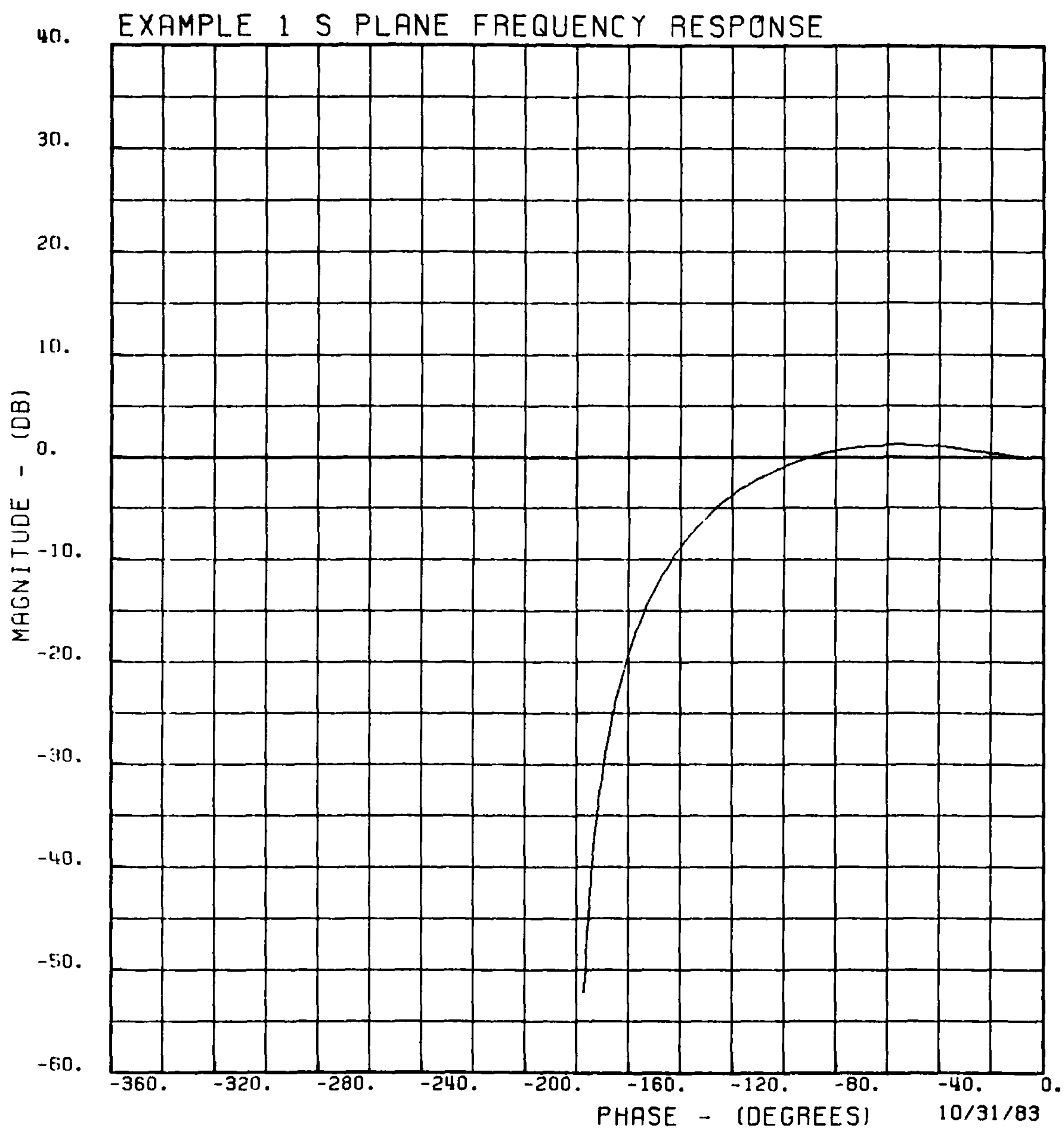
(c) Work has been initiated to convert this program to FORTRAN 5 and have it operational for the IBM 3033 computer as well.

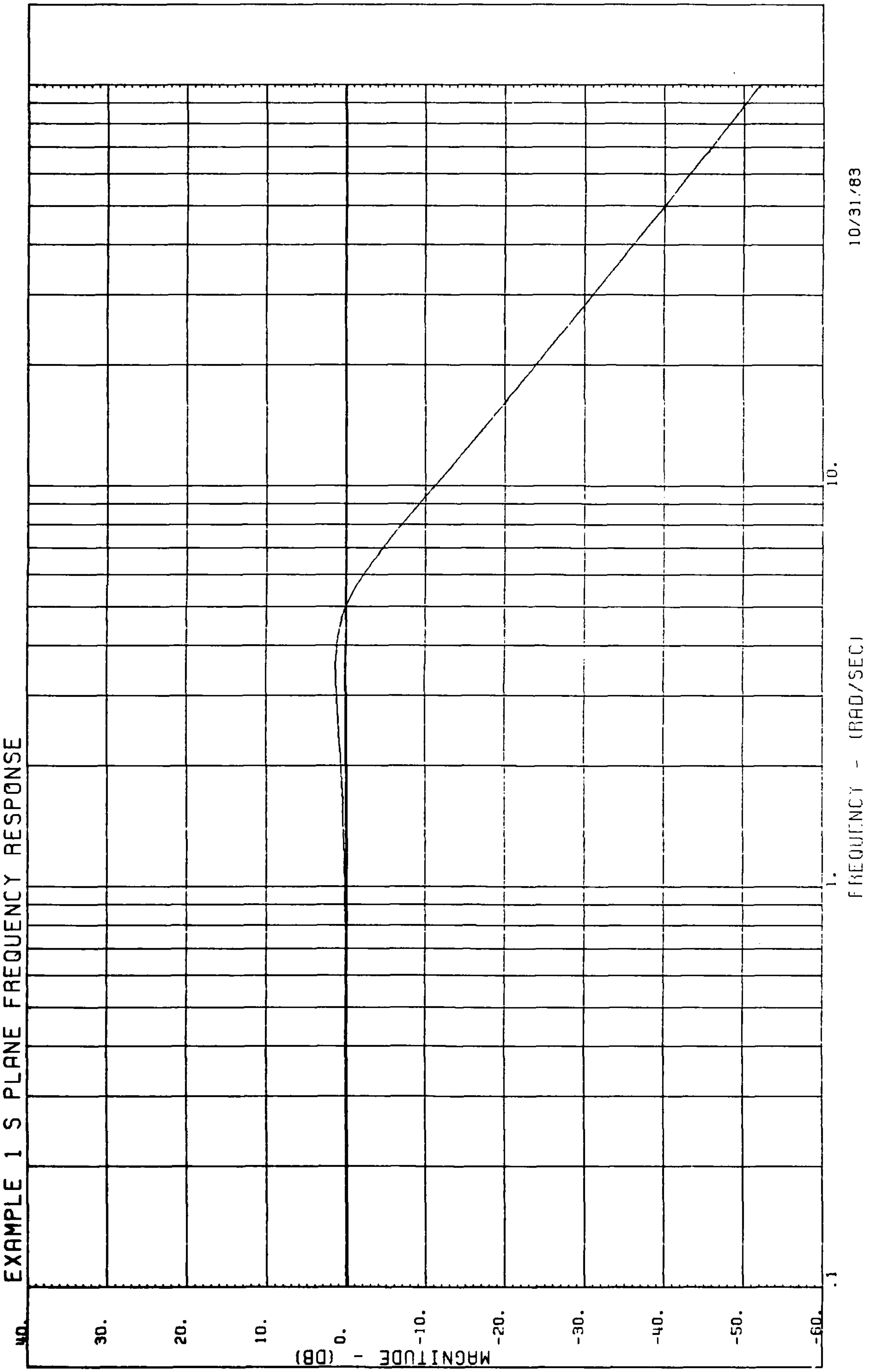
This program as well as this user's guide is in a continuous process of evolution and development. For these reasons, this program and related materials will be made available under the understanding that no warranty, express or implied, is made by the Aerospace Corporation as to the accuracy and functioning of the program and related materials and that no responsibility for program maintenance is implied.

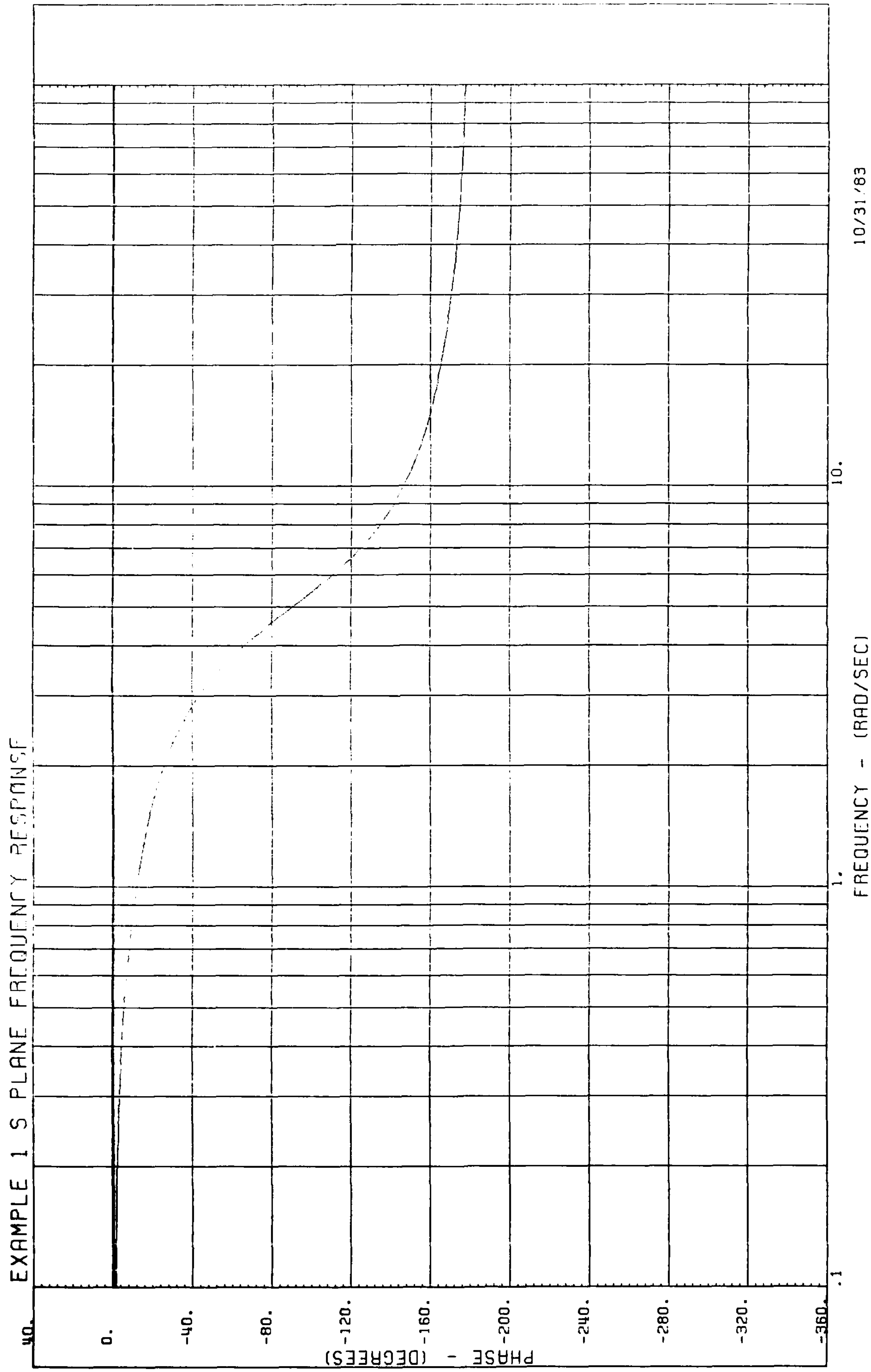
The current reproduction and handling fee is $220.00. Request for a copy of this program, which also includes the Interactive version of LCAP2 as well, should be addressed to:

Administrator
Information Processing Division
The Aerospace Corporation
2350 E. El Segundo Blvd.
El Segundo, California 90245

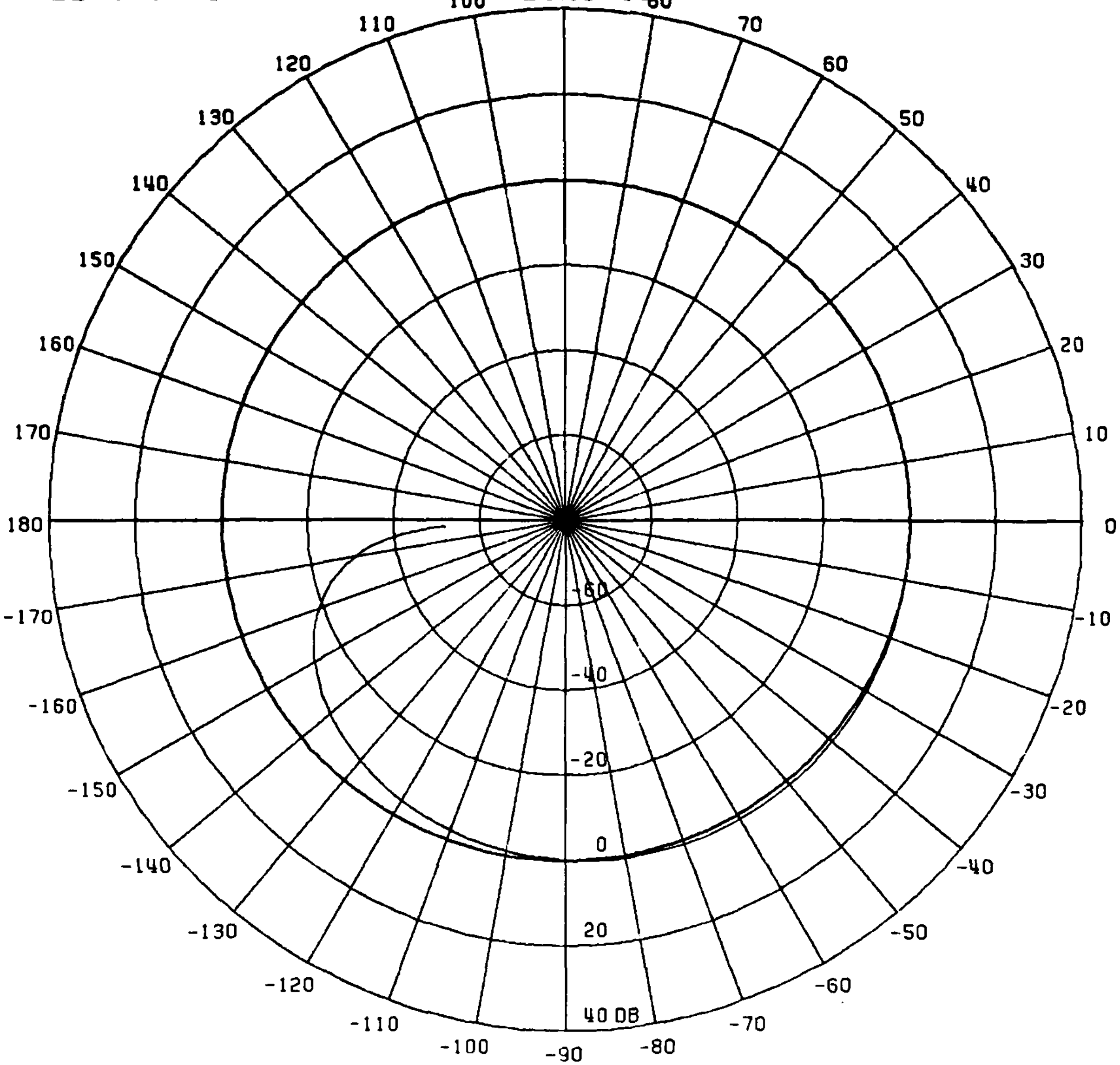
APPENDIX F - HARDCOPY PLOTS FROM EXAMPLES 1,5,6,7,9 AND 10


EXAMPLE 1 S PLANE FREQUENCY RESPONSE
40.
30.
20.
10.
0.
-10.
-20.
-30.
-40.
-50.
-60.
MAGNITUDE - (DB)
-360.
-320.
-280.
-240.
-200.
-160.
-120.
-80.
-40.
0.
PHASE - (DEGREES)
10/31/83


EXAMPLE 1 S PLANE FREQUENCY RESPONSE
MAGNITUDE - (DB)
40.
30.
20.
10.
0.
-10.
-20.
-30.
-40.
-50.
-60.
FREQUENCY - (RAD/SEC)
.1
1.
10.
10/31/83

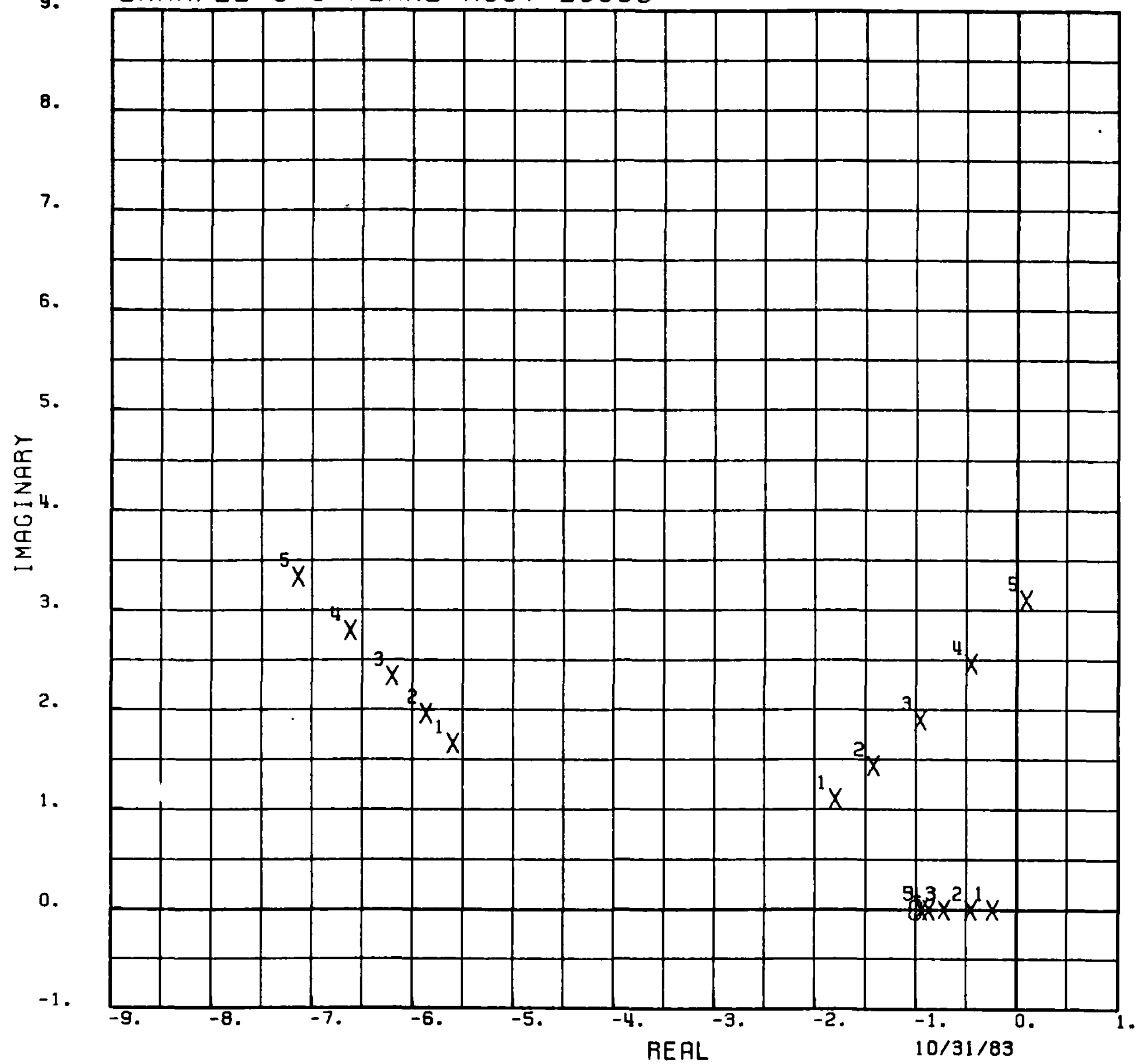

EXAMPLE 1 S PLANE FREQUENCY RESPONSE
10/31/83
FREQUENCY - (RAD/SEC)
.1
1.
10.
PHASE - (DEGREES)
40.
0.
-40.
-80.
-120.
-160.
-200.
-240.
-280.
-320.
-360.

EXAMPLE 1 S PLANE FREQUENCY RESPONSE

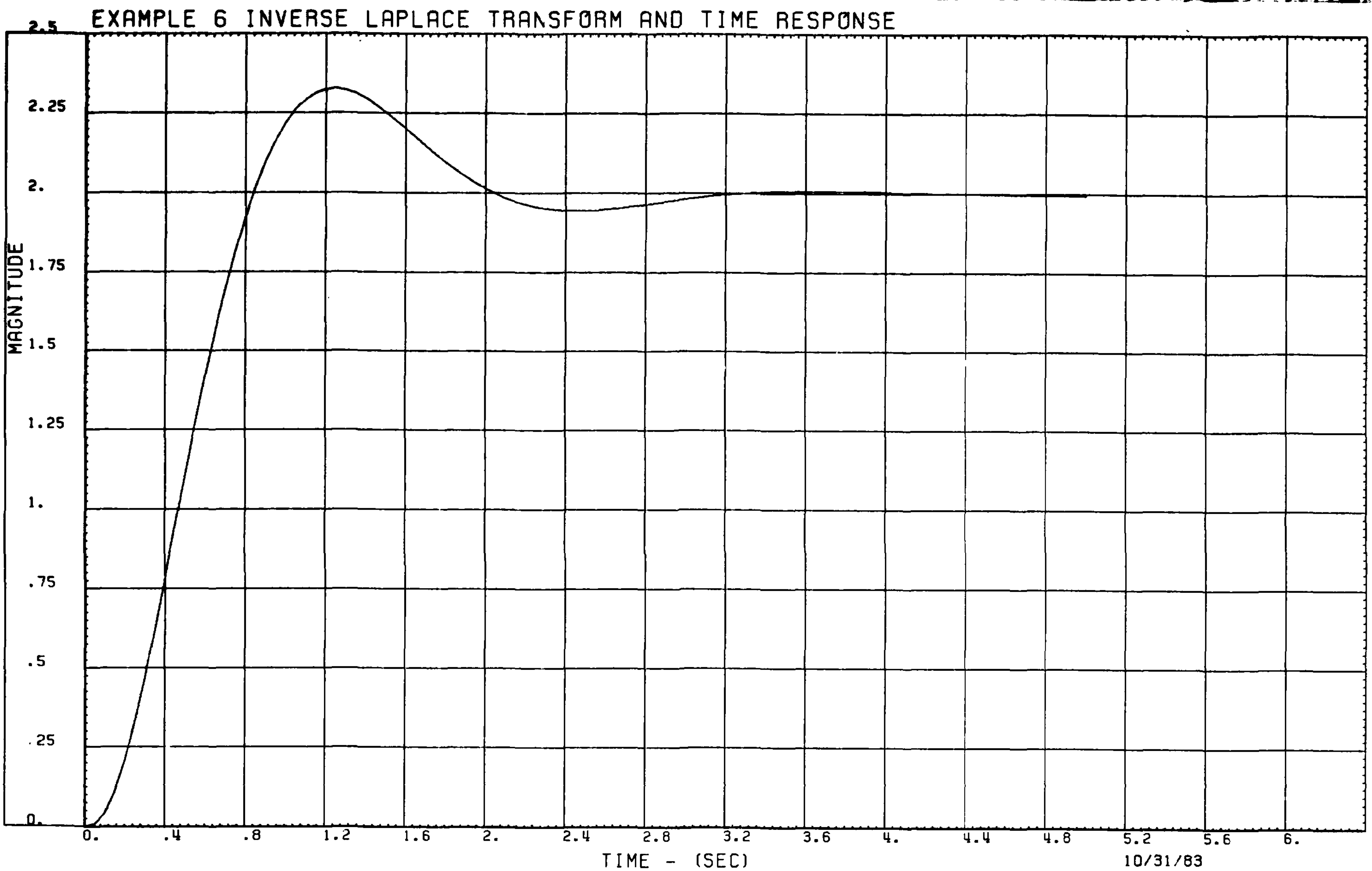


10/31/83

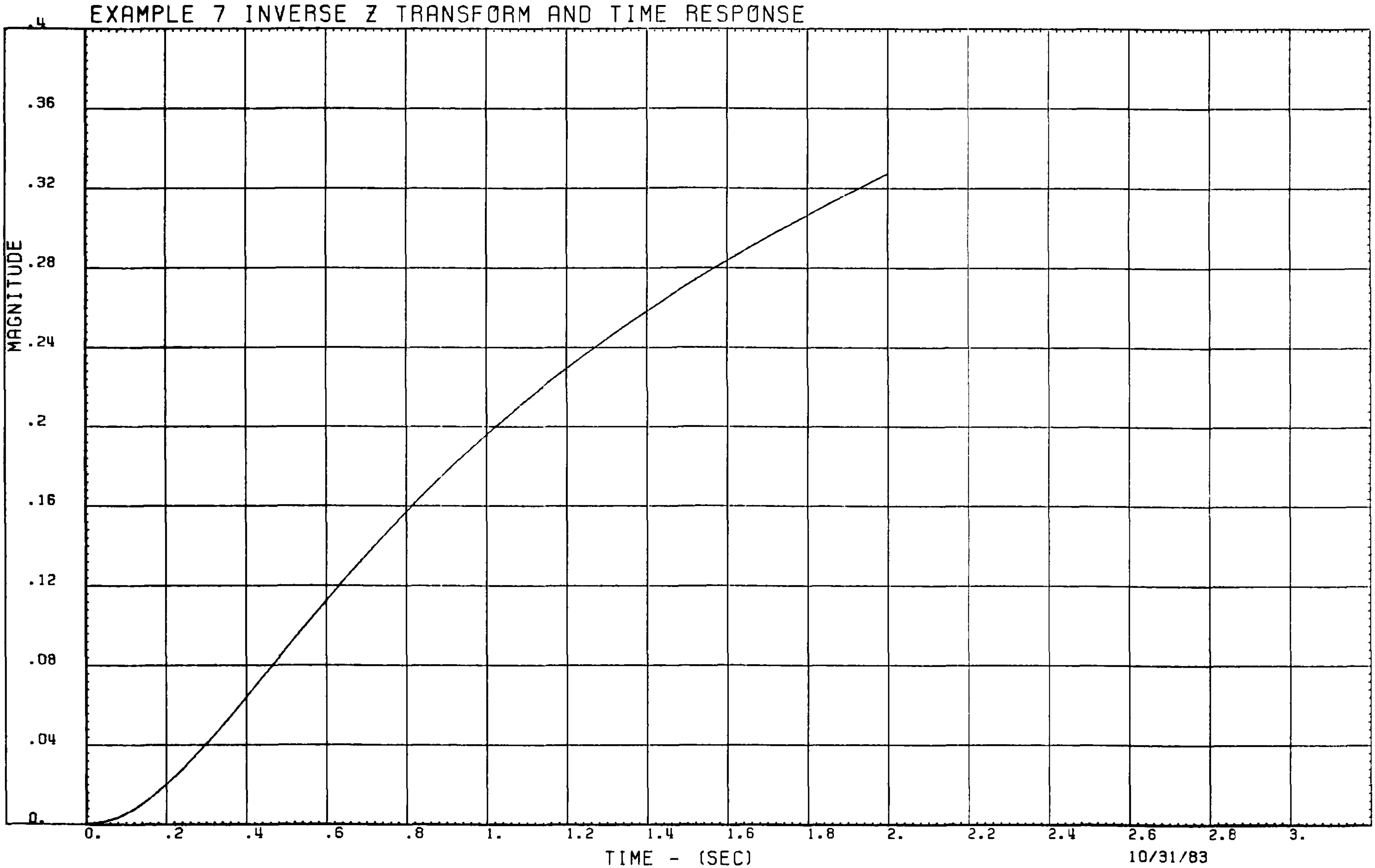
EXAMPLE 5 S PLANE ROOT LOCUS

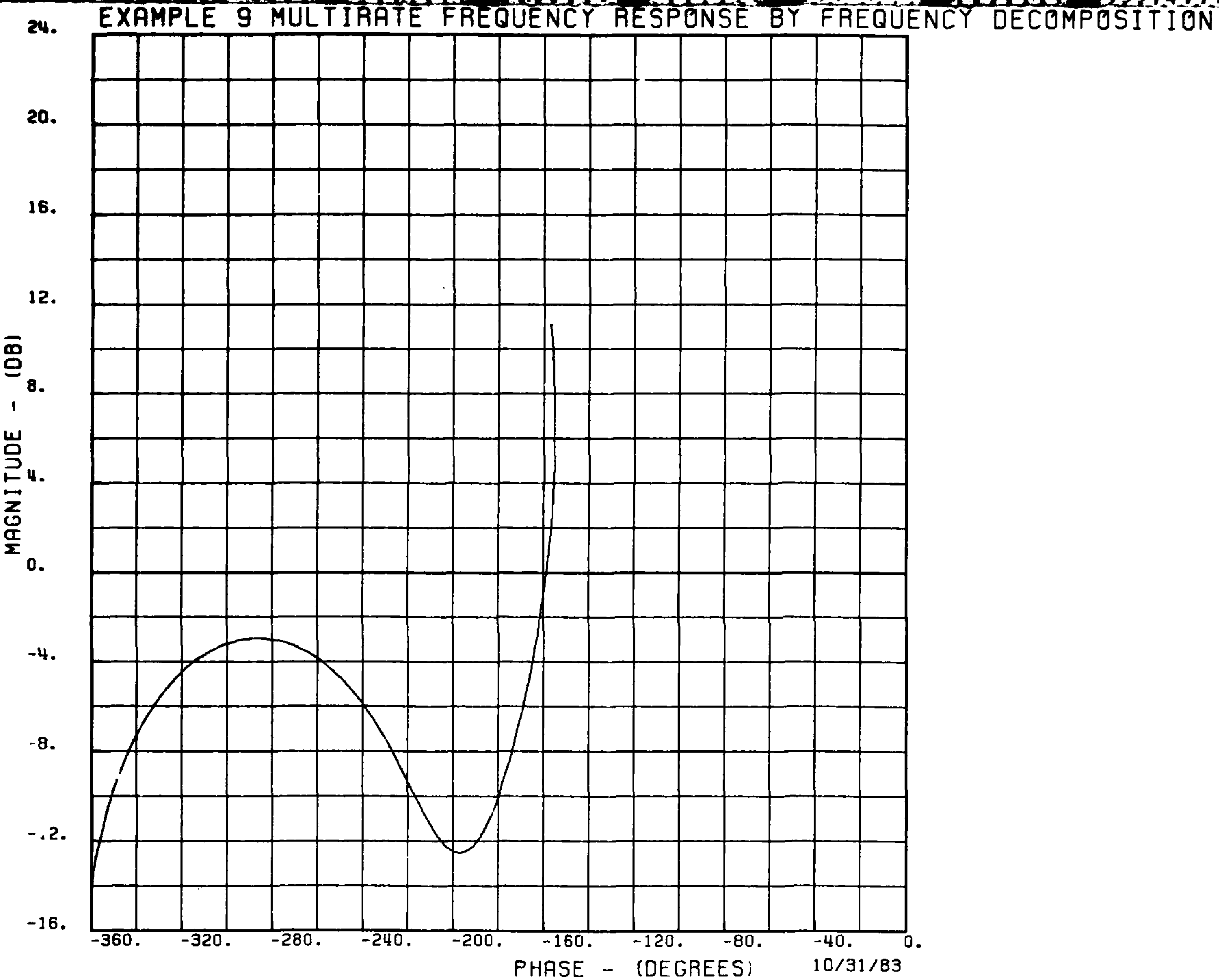


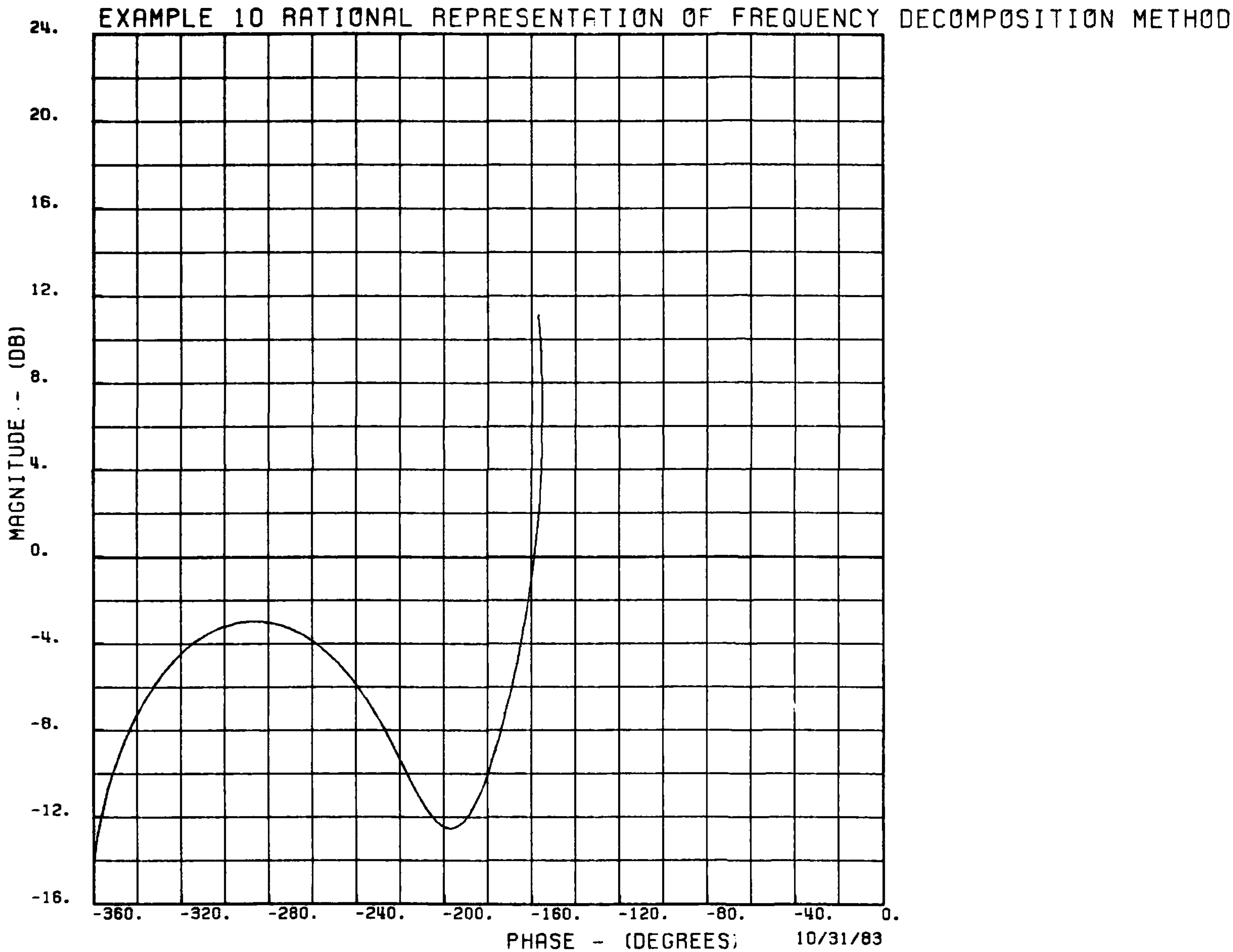
GAIN NO.	GAIN
1	.12500000
2	.25000000
3	.50000000
4	1.0000000
5	2.0000000



EXAMPLE 6 INVERSE LAPLACE TRANSFØRM AND TIME RESPØNSE
MAGNITUDE
2.5
2.25
2.
1.75
1.5
1.25
1.
.75
.5
.25
0.
0.
.4
.8
1.2
1.6
2.
2.4
2.8
3.2
3.6
4.
4.4
4.8
5.2
5.6
6.
TIME - (SEC)
10/31/83


EXAMPLE 7 INVERSE Z TRANSFORM AND TIME RESPONSE
MAGNITUDE
.4
.36
.32
.28
.24
.2
.16
.12
.08
.04
0.
0.
.2
.4
.6
.8
1.
1.2
1.4
1.6
1.8
2.
2.2
2.4
2.6
2.8
3.
TIME - (SEC)
10/31/83


EXAMPLE 9 MULTIRATE FREQUENCY RESPONSE BY FREQUENCY DECOMPOSITION
24.
20.
16.
12.
8.
4.
0.
-4.
-8.
-.2.
-16.
MAGNITUDE - (DB)
-360.
-320.
-280.
-240.
-200.
-160.
-120.
-80.
-40.
0.
PHASE - (DEGREES)
10/31/83



U.S. GOVERNMENT PRINTING OFFICE: 1984-783-004/2127

END

FILMED

DTIC